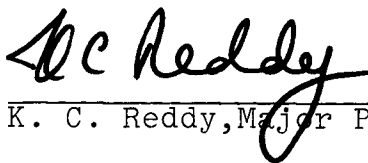
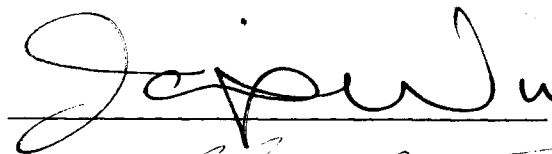
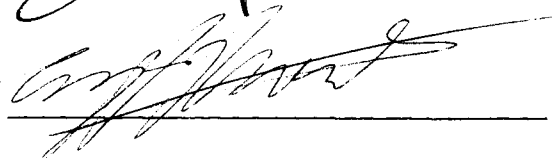


To the Graduate Council:

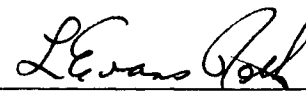
I am submitting herewith a thesis written by Yeng-Yung Tsui entitled "Relaxation Schemes For Inviscid Transonic Flows." I recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Aerospace Engineering.


K. C. Reddy, Major Professor

We have read this thesis
and recommend its acceptance:

Accepted for the Council:


Vice Chancellor
Graduate Studies and Research

RELAXATION SCHEMES FOR INVISCID
TRANSONIC FLOWS

A Thesis

Presented for the

Master of Science

Degree

The University of Tennessee, Knoxville

Yeng-Yung Tsui

March 1981

3049655

ACKNOWLEDGEMENTS

The author wishes to express his gratitude to Dr. K. C. Reddy for his assistance and guidance. This work could never have been done without his help.

Sincere appreciation is also extended to Dr. J. M. Wu for his support and encouragement. The author is also thankful to the committee member, Dr. T. Moulden.

Special thanks are given to Mrs. Melody Ward for her help in typing the final manuscript.

The author is indebted to his parents for their encouragement.

ABSTRACT

A class of numerical schemes for inviscid transonic flows are studied in this thesis. They are based on second order accurate central difference approximations(box-scheme) to the viscous-transonic equation. A successive line relaxation method (SLR), an alternating direction implicit method (ADI) and a multigrid ADI method are developed for solving the non-linear difference equations. Inviscid transonic flows over a symmetric airfoil at zero-angle of attack in an infinite domain and in a wind-tunnel environment with wall suction are computed using the three relaxation schemes. Both subcritical and supercritical flows are calculated. Convergence histories of the approximate solutions show that the ADI method is about three times faster than the line relaxation method and multigrid ADI method is about twice as fast as the ADI method.

TABLE OF CONTENTS

CHAPTER	PAGE
I. INTRODUCTION.	1
II. DIFFERENCE SCHEME AND A SUCCESSIVE LINE RELAXATION METHOD.	4
III. ALTERNATING DIRECTION IMPLICIT METHOD	16
IV. MULTIPLE GRID METHOD.	26
V. RESULTS AND CONCLUSIONS	33
BIBLIOGRAPHY	55
VITA	58

LIST OF FIGURES

FIGURE	PAGE
1. Flow Field and its Boundary Conditions.	7
2. Finite Difference Grid.	10
3. Numerical Flow Field and its Boundary Conditions.	12
4. Upper and Lower Boundaries.	22
5. Correction Storage (CS) Algorithm	29
6. Full Approximation Storage (FAS) Algorithm. . .	32
7. Surface Pressure Coefficient on a 1/6 Thick Circular Airfoil in a Wind Tunnel for $M_\infty = 0.696$	35
8. Convergence History of Residue for $M_\infty = 0.696$ in a Wind Tunnel.	36
9. Convergence History of $\Delta\phi$ for $M_\infty = 0.696$ in a Wind Tunnel.	38
10. Surface Pressure Coefficient on a 1/6 Thick Circular Airfoil in a Wind Tunnel for $M_\infty = 0.791$	41
11. Convergence History of Residue for $M_\infty = 0.791$ in a Wind Tunnel.	42
12. Convergence History of $\Delta\phi$ for $M_\infty = 0.791$ in a Wind Tunnel.	44

FIGURE	PAGE
13. Surface Pressure Coefficient on a 1/6 Thick Circular Airfoil in a Semi-infinite Domain for $M_\infty = 0.791$	47
14. Convergence History of Residue for $M_\infty = 0.791$ in a Semi-infinite Domain.	48
15. Convergence History of $\Delta\phi$ for $M_\infty = 0.791$ in a Semi-infinite Domain.	50

LIST OF SYMBOLS

SYMBOL

C_p	=	pressure coefficient
f^k	=	residue on G^k
$\bar{f}^k$	=	approximate nonhomogeneous term on G^k
$f(x)$	=	body shape function
F	=	slope of body shape $f'(x)$, or nonhomogeneous term of a differential equation (Ch.4)
F^k	=	nonhomogeneous term of a difference equation on G^k
g_j	=	grid size in y-direction
G^k	=	kth grid-level
h_i	=	grid size in x-direction
I_k^{k+1}	=	injection operator
I_{k+1}^k	=	interpolation operator
K_i	=	inviscid-transonic similarity parameter
K_v	=	viscous-transonic similarity parameter
L	=	operator for partial differential equation
$\bar{L}$	=	characteristic length
L_x	=	partial differential operator in x only
L_y	=	partial differential operator in y only
M_x	=	linearized L_x

SYMBOL

M_∞	=	mach number at infinity
P_r''	=	Prandtl number based on $\bar{u}''$
t	=	time
$u^k, \bar{u}^k$	=	approximate solution on G^k
$\bar{u}, \bar{v}$	=	freestream velocity components in x-direction and y-direction, respectively
U	=	small perturbation velocity component in x-direction, or exact solution of a differential equation (Ch.4)
U^k	=	exact solution of a difference equation
$\bar{U}_\infty$	=	freestream velocity at infinity
v	=	small perturbation velocity component in y-direction
v^k	=	correction to u^k
W	=	ϕ_{xx}
x, y	=	nondimensionalized coordinates
$\bar{x}, \bar{y}$	=	cartesian coordinates

Greek Symbols

Δ	=	small increment
ϵ	=	perturbation parameter
γ	=	specific heat ratio
λ	=	stretch factor
$\bar{\mu}$	=	absolute viscosity

SYMBOL

$\bar{\mu}'$	=	bulk viscosity
$\bar{\mu}''$	=	longitudinal viscosity
O	=	order of magnitude
ϕ	=	small perturbation velocity potential
τ	=	nondimensionalized thickness ratio
$\bar{\tau}$	=	airfoil thickness

Subscripts

i,j	=	grid point indices in x-,y-directions
x,y	=	derivatives with respect to x, y
w	=	wall

Superscripts

k	=	on kth grid-level
n	=	time-step index
-	=	physical property (Ch.2), or approximation (Ch.4)
'	=	differentiation
*	=	intermediate property

CHAPTER I

INTRODUCTION

In this thesis we develop finite difference iterative methods to compute inviscid transonic flows past airfoil configurations. A small disturbance equation with a compressive viscosity term in it, known as the viscous-transonic equation, is approximated by a second order accurate central difference scheme. Boundary-layer viscosity is not considered. A system of finite difference equations is set up to approximate the partial differential equation and the boundary conditions. Different relaxation algorithms are adopted to solve this system of equations. A successive line relaxation (SLR) scheme is one such algorithm. An improved algorithm with faster convergence is an alternating direction implicit (ADI) method. Convergence is further speeded up by the use of a multiple grid algorithm together with the ADI method.

In a relaxation scheme for solving a system of finite difference equations the error is composed of a discrete spectrum of wave lengths, which range from the width of the region to the width of a mesh cell. The high frequency components of error are smoothed out quite rapidly in a relaxation procedure, but the low frequency components do not dissipate so quickly. Hence, the convergence rate be-

comes asymptotically very slow. The idea of the multiple grid method is based on the fact that after several iterations on fine grids when the high frequency errors diminish, the relaxation sweeps are moved to a coarse grid. A "residual" equation is then solved on the coarse mesh. The resulting correction is added to the last approximation on the fine mesh. Similar procedure can be repeated to approximate solutions of "residual" equation on even coarser grids. It follows that the solution process involves three main steps:

- (1) relaxation sweeps on each level
- (2) fine-grid-to-coarse-grid transfers of residuals
- (3) coarse-grid-to-fine-grid interpolations of corrections.

This is the "correction storage (CS)" method and is only suitable to linear problems. For nonlinear cases a modified method called "full approximation storage (FAS)" scheme is adopted. In this method, instead of storing the corrections, the full current approximations are stored and, instead of the residual equation, a corrected equation which is an approximation to the difference equation on fine grids is solved on coarse grids. Then the same procedure is followed as in the CS algorithm.

The success of multiple grid method depends on the use of a relaxation algorithm, which rapidly reduces

the high frequency errors. Since point and line relaxation schemes do not reduce these errors so quickly in a general stretched mesh, where the mesh aspect ratio varies from very small to very large values [1,2], it is proposed to use an alternating direction implicit (ADI) method as the iteration algorithm.

Since an iteration method can be regarded as a time-dependent relaxation scheme where time is a relaxation parameter. A time-dependent term can be added to the viscous-transonic flow equation. Then the ADI method is applied to this time-dependent relaxation equation. The ADI method is a two-step method in which horizontal and vertical line relaxation sweeps in x and y directions are carried out alternately for a two-dimensional flow problem. The intermediate solution after each of the horizontal line relaxation sweeps is corrected by the vertical line relaxations. Thus the final result after these two partial steps is an approximate solution for one time-step. This process is continued till the time-dependent term approaches zero.

Three kinds of methods will be developed in this thesis: successive horizontal line relaxation method, ADI method, and multiple grid method. The convergent rates of these different methods will be compared by using the residue norm and $\Delta\phi$ norm for several test problems dealing with subcritical flows and supercritical flows in a semi-infinite domain with free flow far field boundary conditions and in a confined domain in a wind tunnel.

CHAPTER II

DIFFERENCE SCHEME AND A SUCCESSIVE
LINE RELAXATION METHOD

The inviscid-transonic equation

$$(K-(\gamma+1)\phi_x) \phi_{xx} + \phi_{yy} = 0 \quad (1)$$

which is the small disturbance approximation of the compressible potential flow equations, is a mixed elliptic-hyperbolic type. Mixed difference schemes are used extensively to solve such an equation. Murman and Cole [3,4] have originated this approach and Jameson [5] has developed more sophisticated versions of this scheme. In this method the velocity is tested to determine whether the flow is subsonic or supersonic at each point during an iteration process. In the supersonic region a central difference scheme is used to approximate the derivatives. This implies that the domain of dependence of an elliptic equation extends in all directions. In supersonic region where the equation is hyperbolic, a backward difference scheme is employed in view of the fact that the domain of dependence is in the upwind direction. Special parabolic and shock-point operators are used at points where the flow enters and leaves the supersonic zone, respectively. Because of the different orders of truncation errors

introduced by the central difference and backward difference schemes, artificial viscosities are induced in the supersonic region.

The viscous transonic (V-T) equation is a small disturbance approximation of the Navier-Stokes equations, and includes the effects of compressive viscosity across shock regions but not the effects of boundary layer viscosity. It is originally derived by Liepmann, Ashkenas, and Cole (1948). Sichel has given its derivation and reviewed its properties and applications in a review article [6]. The viscous-transonic equation for a flow past a symmetric airfoil given below is in Sichel's notation.

$$K_V \phi_{xxx} + (K_i - \phi_x) \phi_{xx} + \phi_{yy} = 0 \quad (2)$$

where

$$K_V = \frac{\bar{u}''}{\bar{\rho}_\infty \bar{U}_\infty \bar{L} \tau^{2/3} M_\infty^{4/3} (\gamma+1)^{2/3}} \left(1 + \frac{\gamma-1}{P_r} \right)$$

is the viscous-transonic similarity parameter,

$$K_i = \frac{1 - M_\infty^2}{\tau^{2/3} M_\infty^{4/3} (\gamma+1)^{2/3}}$$

is the usual inviscid transonic similarity parameter,
and

ϕ is the perturbation velocity potential,

$\bar{L}$ is the characteristic length,

$$x = \bar{x}/\bar{L} ,$$

$$y = \lambda \bar{y}/\bar{L} ,$$

$\tau = \bar{\tau}/\bar{L}$ is the thickness ratio,

$$t = \tau^{2/3}/(\gamma+1)^{1/3} M_\infty^{2/3} ,$$

$$\lambda = \epsilon/\tau ,$$

$\bar{\mu}'' = \frac{4}{3} \bar{\mu} + \bar{\mu}'$ is the longitudinal viscosity,

P_r'' is a prandtl number based on $\bar{\mu}''$,

$$\frac{\bar{u}}{\bar{u}_\infty} = 1 + \epsilon U \quad \text{where } U = \phi_x ,$$

$$\frac{\bar{v}}{\bar{u}_\infty} = \lambda \epsilon v \quad \text{where } v = \phi_y .$$

The boundary conditions (Fig.1) are

$$\phi_y(x,0) = f'(x) \quad (3-a)$$

where $f'(x)$ is the slope of the body shape, and

$$\phi \rightarrow 0 \quad \text{as } r = \sqrt{x^2 + y^2} \rightarrow \infty \quad (3-b)$$

The pressure coefficient is

$$C_p = -2\epsilon \phi_x = \frac{-2\tau^{2/3}}{(\gamma+1)^{1/3} M_\infty^{2/3}} \phi_x . \quad (4)$$

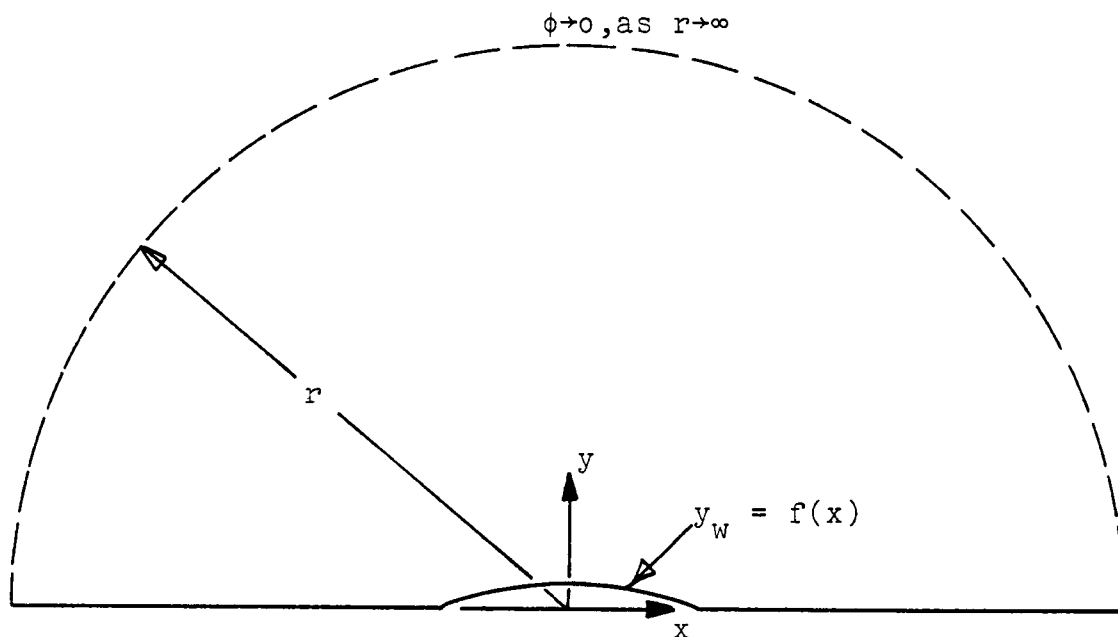


Figure 1. Flow field and its Boundary Conditions.

The viscosity term $K_v \phi_{xxx}$ is the contribution of the compressive viscosity which is prominent across a shock. The parameter K_v is rather small for high Reynolds number flows. A small K_v makes the equation extremely stiff. The equation, however, is parabolic of higher order everywhere and thus may not require mixed differencing in principle. For subsonic far field, the equation would still require boundary conditions all around the domain of computation like an elliptic problem. The size of K_v affects mainly the shock thickness. In numerical computations extremely small K_v makes the calculations ill-conditioned. We use a moderately small K_v which can still produce a thin shock region without making the system

overly stiff. Thus $K_v \phi_{xxx}$ term, in a sense, plays the role of an artificial viscosity term in the numerical calculations. This approach, however, as will be seen later, would help preserve second order accuracy for all the flow variables in the numerical scheme.

Keller has developed a "box scheme" to solve a large class of parabolic problems [7]. The following method was developed by Reddy [8] to solve the viscous-transonic equation with the application of the box scheme.

Let $U = \phi_x$, $V = \phi_y$, $W = \phi_{xx}$, and the viscous-transonic equation can be rewritten as a system of first-order partial differential equations.

$$\phi_x = U \quad , \quad (5-a)$$

$$U_x = W \quad , \quad (5-b)$$

$$(K_v W + K_1 U - \frac{1}{2} U^2)_x = -V_y \quad . \quad (5-c)$$

It may be noted that the third equation which represents the V-T equation is in conservation form.

In problems where shocks are expected, we are really looking for the so called weak solutions of the system. In this problem while the shock region is not a discontinuity, it is a region of sharp gradients approaching a discontinuity. Difference approximations of the

conservation form of the p.d.e. approximate the integral conservation laws and thus can yield weak solutions with near discontinuities, if they exist, with proper shock jumps.

The finite difference approximations to the first order system, equations(5), are written at the point $(x_{i-\frac{1}{2}}, y_j)$, where i, j are the grid point indices (Fig.2) in the x, y directions.

$$D_x \phi_{i-\frac{1}{2}, j} = U_{i-\frac{1}{2}, j} \quad , \quad (6-a)$$

$$D_x U_{i-\frac{1}{2}, j} = W_{i-\frac{1}{2}, j} \quad , \quad (6-b)$$

$$D_x (K_v W + K_1 U - \frac{1}{2} U^2)_{i-\frac{1}{2}, j} = -D_y V_{i-\frac{1}{2}, j} \quad . \quad (6-c)$$

where $D_x A_{i-\frac{1}{2}, j} = (A_{1j} - A_{i-1j})/h_i \quad ,$

$$D_y V_{i-\frac{1}{2}, j} = 2(V_{i-\frac{1}{2}, j+\frac{1}{2}} - V_{i-\frac{1}{2}, j-\frac{1}{2}})/(g_j + g_{j+1}) \quad ,$$

and A can be ϕ , U , and $(K_v W + K_1 U - \frac{1}{2} U^2)$,

$$A_{i-\frac{1}{2}, j} = (A_{1j} + A_{i-1j})/2 \quad ,$$

$$V_{i-\frac{1}{2}, j+\frac{1}{2}} = D_y \phi_{i-\frac{1}{2}, j+\frac{1}{2}} = (\phi_{i-\frac{1}{2}, j+1} - \phi_{i-\frac{1}{2}, j})/g_{j+1} \quad ,$$

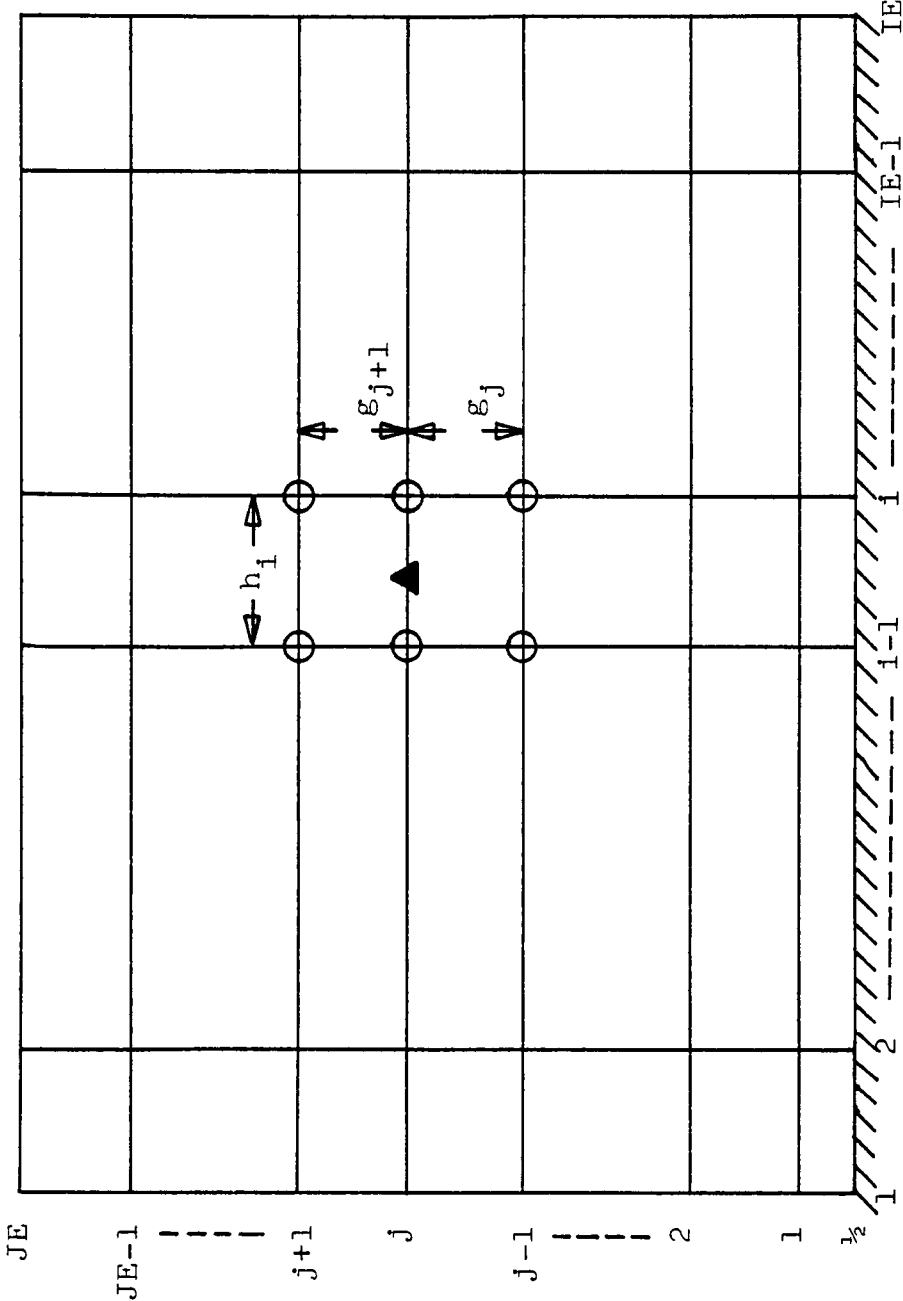


Figure 2. Finite Difference Grid.

$$V_{i-\frac{1}{2},j-\frac{1}{2}} = D_y \phi_{i-\frac{1}{2},j-\frac{1}{2}} = (\phi_{i-\frac{1}{2},j} - \phi_{i-\frac{1}{2},j-1})/g_j ,$$

$$h_i = x_i - x_{i-1} ,$$

$$g_j = y_j - y_{j-1} .$$

In this method since we approximate the first order system by a second order difference scheme, the unknowns are ϕ , ϕ_x , and ϕ_{xx} at all grid points and these are computed simultaneously to second order accuracy.

The boundary conditions (Fig. 3) are approximated as

$$(i) \quad V_{i,\frac{1}{2}} = \begin{cases} \left(\frac{df}{dx}\right)_i \equiv F_i & ; |x_i| \leq 1 \text{ at } j=\frac{1}{2} , \\ 0 & ; |x_i| > 1 \end{cases} \quad (7-a)$$

$$(ii) \quad \phi = 0 \quad \text{at } j=JE , \quad (7-b)$$

$$(iii) \quad \phi = 0 \quad \text{at } i = 1 , \quad (7-c)$$

$$(iv) \quad \phi_{xx} = W = 0 \quad \text{at } i = IE . \quad (7-d)$$

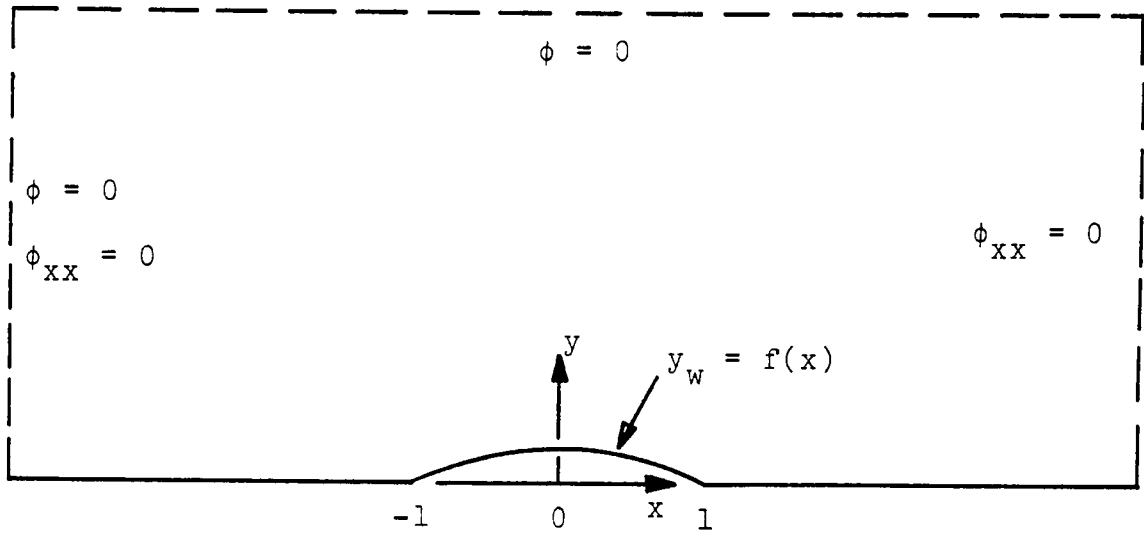


Figure 3. Numerical Flow Field and its Boundary Conditions.

The asymptotic solution of ϕ has the property: $\phi \approx O(\frac{1}{r})$ for large r and thus the derivatives of $\phi \rightarrow 0$ faster than ϕ . Hence in the numerical computations where the semi-infinite domain is replaced by a finite region, it is advantageous to replace $\phi = 0$ by $\phi_{xx} = 0$ at the right boundary as shown above.

The nonlinear system of equations for ϕ , U , and W at the grid points are solved by a successive line relaxation (SLR) method. Solutions on each horizontal line are relaxed in the y -direction successively. The latest values of P , U , and W are used in every relaxation.

First the equations (4) are linearized on each horizontal line j by Newton's method.

$$A_{i,j} = A_{i,j} + \Delta A_{i,j}$$

$$A_{i-1,j} = A_{i-1,j} + \Delta A_{i-1,j}$$

where A can be ϕ , U , and W . Inserting these into the equations (4) and ignoring the quadratic terms in ΔA , we obtain

$$(\Delta\phi_{i,j} - \Delta\phi_{i-1,j}) - \frac{h_i}{2} (\Delta U_{i,j} + \Delta U_{i-1,j}) = R1 \quad (8-a)$$

$$\text{where } R1 = -h_i (D_x \phi_{i-\frac{1}{2},j} - U_{i-\frac{1}{2},j}) ,$$

$$(\Delta U_{i,j} - \Delta U_{i-1,j}) - \frac{h_i}{2} (\Delta W_{i,j} + \Delta W_{i-1,j}) = R2 \quad (8-b)$$

$$\text{where } R2 = -h_i (D_x U_{i-\frac{1}{2},j} - W_{i-\frac{1}{2},j}) ,$$

$$\text{and } a1_{i,j} \Delta\phi_{i-1,j} + a2_{i,j} \Delta U_{i-1,j} + a3_{i,j} \Delta W_{i-1,j} +$$

$$b1_{i,j} \Delta\phi_{i,j} + b2_{i,j} \Delta U_{i,j} + b3_{i,j} \Delta W_{i,j} = R3 \quad (8-c)$$

$$\text{where } a1_{i,j} = \begin{cases} -\frac{h_i}{g_2(g_1+g_2)} & \text{at } j=1 , \\ -\frac{h_i}{g_j g_{j+1}} & \text{at } j \geq 2 , \end{cases}$$

$$a_{2,i,j} = -K_i + U_{i-1,j} ,$$

$$a_{3,i,j} = -K_v ,$$

$$b_{1,i,j} = a_{1,i,j} ,$$

$$b_{2,i,j} = K_i - U_{i,j} ,$$

$$b_{3,i,j} = K_v ,$$

$$\text{and } R3 = -h_1 \left[D_x (K_v W + K_i U - \frac{1}{2} U^2)_{i-\frac{1}{2},j} + RY \right] ,$$

$$\text{where } RY = \begin{cases} \frac{2}{g_j + g_{j+1}} \left[\frac{\phi_{i-\frac{1}{2},j+1} - \phi_{i-\frac{1}{2},j}}{g_{j+1}} - \frac{\phi_{i-\frac{1}{2},j} - \phi_{i-\frac{1}{2},j-1}}{g_j} \right] & \text{at } j \geq 2 , \\ \frac{2}{g_1 + g_2} \left[\frac{\phi_{i-\frac{1}{2},2} - \phi_{i-\frac{1}{2},1}}{g_2} - F_{i-\frac{1}{2}} \right] & \text{at } j=1 , \end{cases}$$

$$\text{where } F_{i-\frac{1}{2}} \equiv \left(\frac{df}{dx} \right)_{i-\frac{1}{2}} .$$

Boundary conditions become

$$\Delta \phi_{1,j} = 0 , \quad (9-a)$$

$$\Delta W_{1,j} = 0 , \quad (9-b)$$

$$\Delta W_{IE,j} = 0 \quad (9-c)$$

The coefficient matrix of this linear system of equations is in block-tridiagonal form. Thus $\Delta\phi$, ΔU , and ΔW can be solved on the j th horizontal line. The iterations can be repeated on each line for the Newton iteration to converge. As the calculation proceeds for a while, it does not take more than two or three iterations for the Newton iteration to converge on each line. However, since several hundred sweeps of the whole region are necessary for the line relaxation method to converge, it is really not necessary to carry out more than one Newton iteration on each line.

CHAPTER III

ALTERNATING DIRECTION IMPLICIT METHOD

Alternating direction implicit (ADI) method is one in which a horizontal line relaxation and a vertical line relaxation are alternated in every sweep. This method has the potential of reducing the errors faster than the line relaxation methods. Peaceman and Rachford [9] have first developed an ADI method for solving multi-dimensional heat transfer and Laplace equation problems. There is a vast amount of literature on the application of various kinds of ADI methods to time dependent problems or relaxation problems. ADI methods can also be interpreted as time-splitting methods and approximate factorization methods. Jameson [10] has used ADI to solve 2D and 3D transonic flow problems. Beam and Warming [11] use the approximate factorization and ADI techniques to solve the time dependent Navier-Stokes equations.

To develop an ADI method for the V-T equation, we introduce a time derivative, where time is really a relaxation parameter. Let

$$\begin{aligned}
 \frac{\partial \phi}{\partial t} &= (K_v W + K_1 U - \frac{1}{2} U^2)_x + V_y \\
 &= L_x \phi + L_y \phi \\
 &= L \phi
 \end{aligned}
 \tag{10}$$

where $L_x \phi = (K_v \phi_{xx} + K_1 \phi_x - \frac{1}{2} \phi_x^2)_x = (K_v W + K_1 U - \frac{1}{2} U^2)_x$,

and $L_y \phi = V_y = \phi_{yy}$.

It may be noted that we intend to make $\frac{\partial \phi}{\partial t} \rightarrow 0$ or $L\phi \rightarrow 0$ as t becomes sufficiently large.

Let the temporal variable t be discretized as $t = n\Delta t$ where Δt is the discrete time increment and $\phi(t) = \phi(n\Delta t) = \phi^n$. The equation (10) is approximated at the time-step $(n+\frac{1}{2})\Delta t$ by

$$\left(\frac{\partial \phi}{\partial t}\right)^{n+\frac{1}{2}} = \frac{1}{2} \left[\left(\frac{\partial \phi}{\partial t}\right)^n + \left(\frac{\partial \phi}{\partial t}\right)^{n+1} \right] + O(\Delta t^2) \quad (11)$$

By Taylor series expansions, we can write

$$\begin{aligned} \left(\frac{\partial \phi}{\partial t}\right)^{n+1} &= \left(\frac{\partial \phi}{\partial t}\right)^n + \frac{\partial}{\partial t} \left(\frac{\partial \phi}{\partial t}\right)^n \Delta t + O(\Delta t^2) \\ &= \left(\frac{\partial \phi}{\partial t}\right)^n + \frac{\Delta}{\Delta t} \left(\frac{\partial \phi}{\partial t}\right)^n \Delta t + O(\Delta t^2) \quad , \end{aligned} \quad (12)$$

$$\begin{aligned} \left(\frac{\partial \phi}{\partial t}\right)^{n+\frac{1}{2}} &= \frac{\phi^{n+1} - \phi^n}{\Delta t} + O(\Delta t^2) \\ &= \frac{\Delta \phi^n}{\Delta t} + O(\Delta t^2) \quad . \end{aligned} \quad (13)$$

where $\Delta \phi^n = \phi^{n+1} - \phi^n$.

By inserting equations (12) and (13) into (11) we obtain

$$\Delta\phi^n = \Delta t \left(\frac{\partial\phi}{\partial t} \right)^n + \frac{\Delta t}{2} \Delta \left(\frac{\partial\phi}{\partial t} \right)^n + O(\Delta t^3) \quad (14)$$

and substituting the equation (10) into (14) leads to

$$\Delta\phi^n - \frac{\Delta t}{2} \Delta(L\phi^n) = \Delta t L\phi^n + O(\Delta t^3) \quad (15)$$

where $\Delta L\phi^n = \Delta L_x\phi^n + \Delta L_y\phi^n$ and $\Delta(L_y\phi^n) = L_y\Delta\phi^n$,

since L_y is a linear operator. Since L_x operator has a non-linear term, we apply Taylor series to linearize the non-linear term to give

$$\begin{aligned} \Delta(\phi_x^2)^n &= (\phi_x^2)^{n+1} - (\phi_x^2)^n \\ &= (U^2)^{n+1} - (U^2)^n \\ &= \left[(U^2)^n + \frac{\partial(U^2)^n}{\partial t} \Delta t + O(\Delta t^2) \right] - (U^2)^n \\ &= 2U^n \Delta U^n + O(\Delta t^2) \quad . \end{aligned} \quad (16)$$

The error due to linearization of the non-linear term is of the same order as that of the time truncation error and would not be detrimental to the basic scheme. Hence, a linear operator M_x can be defined as

$$M_x \Delta \phi^n = (K_v \Delta W^n - K_1 \Delta U^n - U^n \Delta U^n)_x \quad . \quad (17)$$

Replacing $\Delta(L_x \phi^n)$ by $M_x \Delta \phi^n$, equation (15) can be factorized as

$$(I - \frac{\Delta t}{2} M_x)(I - \frac{\Delta t}{2} L_y) \Delta \phi^n = \Delta t L \phi^n + O(\Delta t^3) \quad . \quad (18)$$

By ignoring $\frac{\Delta t^2}{4} M_x L_y \Delta \phi^n = \frac{\Delta t^3}{4} M_x L_y \frac{\Delta \phi^n}{\Delta t} = \frac{\Delta t^3}{4} M_x L_y \frac{\partial \phi^n}{\partial t}$.

Which is $O(\Delta t^3)$. Thus the formal accuracy of the equation (15) is not upset by the approximate factorization scheme of the equation (18).

The factored scheme (18) can be implemented by the alternating direction sequence:

$$(I - \frac{\Delta t}{2} M_x) \Delta \phi^* = \Delta t L \phi^n \quad (19-a)$$

$$(I - \frac{\Delta t}{2} L_y) \Delta \phi^n = \Delta \phi^* \quad (19-b)$$

where $\Delta \phi^*$ is an intermediate quantity in the ADI scheme and has no real significance.

Two points should be noted:

(i) If $\phi^{n+1} = \phi^n$, i.e., $\Delta \phi^n = 0$, the factored equation (18) will reduce to $L \phi^n = 0$ which is the original

steady flow equation. It means that the equation (18) represents an iterative method for solving the steady flow.

(ii) The time increment is nothing but an iteration parameter. It can be varied from iteration to iteration in a cyclic manner to speed up the convergence of the scheme.

The first step, equation (19-a) can be rewritten as

$$\Delta\phi^* - \frac{\Delta t}{2} [K_v \Delta W^* + (K_1 - U^n) \Delta U^*]_x = \Delta t L \phi^n \quad (20)$$

which can be approximated at point $(i-\frac{1}{2}, j)$ by a second-order central difference box scheme as before and multiplied by $-h_1/\Delta t$ on both sides to obtain

$$\begin{aligned} & a1_{i,j} \Delta\phi_{i-1,j}^* + a2_{i,j} \Delta U_{i-1,j}^* + a3_{i,j} \Delta W_{i-1,j}^* \\ & + b1_{i,j} \Delta\phi_{i,j}^* + b2_{i,j} \Delta U_{i,j}^* + b3_{i,j} \Delta W_{i,j}^* = R3 \end{aligned} \quad (21)$$

where $a1_{i,j} = -h_1/2\Delta t$,

$$a2_{i,j} = -\frac{1}{2}(K_1 - U_{i-1,j}) \quad ,$$

$$a3_{i,j} = -\frac{1}{2}K_v \quad ,$$

$$b1_{i,j} = a1_{i,j} \quad ,$$

$$b2_{i,j} = \frac{1}{2}(K_i - U_{i,j}) ,$$

$$b3_{i,j} = -a3_{i,j} ,$$

$$\text{and} \quad R3 = -h_i L \phi_{i-\frac{1}{2},j}^n .$$

Together with the difference schemes for ϕ_x , U_x , i.e., equations (8-a), (8-b), and the boundary conditions (9) we obtain a block-tridiagonal system of equations which are similar to the one obtained in the preceding chapter except for the coefficients $a1$, $a2$, ---, $b3$ which are different.

In the vertical sweeps, i.e. the second time step (19-b), we put the lower boundary on line $j=1$ (Fig.4-a) instead of on line $j=\frac{1}{2}$ for our future convenience in multiple grid method and change the upper boundary condition to accomodate the porosity of a wind tunnel wall at that boundary or a free flow condition as before.

The lower boundary condition $\phi_y = \frac{df}{dx} = F(x)$ on line $j=1$ can be reduced to

$$\phi_{i-\frac{1}{2},0} = \phi_{i-\frac{1}{2},2} - 2g_2 F_{i-\frac{1}{2}} \quad (22)$$

by using a central difference for ϕ_y and assuming $g_1=g_2$ where line $j=0$ is only an imaginary line. Thus we obtain

$$(\phi_{yy})_{i-\frac{1}{2},1} = \frac{2(\phi_{i-\frac{1}{2},2} - \phi_{i-\frac{1}{2},1})}{g_2^2} - \frac{2F_{i-\frac{1}{2}}}{g_2} \quad (23)$$

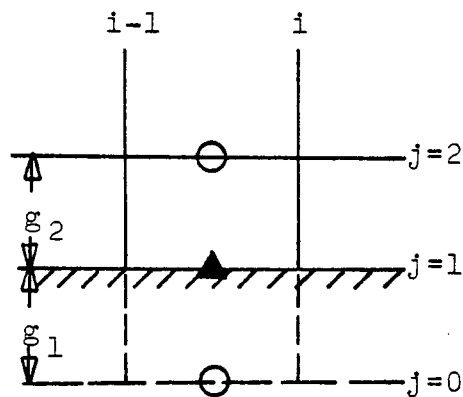
by using a central difference for ϕ_{yy} and equation (22).

The R3 of equation (21) on line $j=1$ becomes

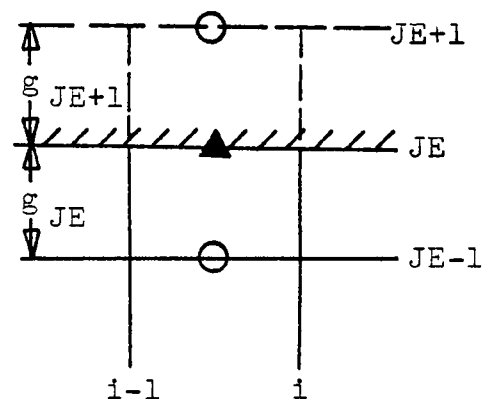
$$R3 = -h_1 L \phi_{i-\frac{1}{2},1}^n = -h_1 \left[(K_v W^n + K_1 U^{n-\frac{1}{2}} U^{n^2})_{i-\frac{1}{2},1} + RY \right] \quad (24)$$

where $RY = (\phi_{yy})_{i-\frac{1}{2},1}$ given by equation (23).

The upper boundary condition is changed to $\phi_y = \alpha(x)$ where $\alpha(x)$ at discrete points along the line $j=JE$ (Fig. 4-b) may be measured by experiments in a wind tunnel. In a free flow problem where the upper boundary



(a). Lower boundary.



(b). Upper boundary.

Figure 4. Upper and Lower Boundaries.

is far away from the airfoil, we set $\alpha(x) = 0$ which amounts to $\phi_y = 0$. This is just as valid as $\phi = 0$ at that boundary for free flow problems. Similarly we can derive

$$(\phi_{yy})_{i-\frac{1}{2},JE} = \frac{2(\phi_{i-\frac{1}{2},JE-1} - \phi_{i-\frac{1}{2},JE})}{g_{JE}^2} + \frac{2\alpha_{i-\frac{1}{2}}}{g_{JE}}. \quad (25)$$

In this case R3 on top line can be obtained from (24) by substituting (25) into RY and replacing $j=1$ by $j=JEND$.

The second step equation (19-b) is approximated at (i,j) by

$$b\Delta\phi_{i,j-1}^n + a\Delta\phi_{i,j}^n + c\Delta\phi_{i,j+1}^n = \Delta\phi_{i,j}^* \quad (26)$$

where $b = -\frac{\Delta t}{g_j(g_j + g_{j+1})},$

$$a = 1 + \frac{\Delta t}{g_{j+1}(g_j + g_{j+1})} + \frac{\Delta t}{g_j(g_j + g_{j+1})}$$

$$c = -\frac{\Delta t}{g_{j+1}(g_j + g_{j+1})} \quad \text{for } 2 \leq j \leq JE-1,$$

and $\Delta\phi_{i,j}^*$ is known from the horizontal sweep.

Boundary conditions for $\Delta\phi$ in the vertical sweep are

$$\Delta V_{i,1} = (\Delta \phi_y)_{i,1} = 0 \quad (27-a)$$

$$\Delta V_{i,JE} = (\Delta \phi_y)_{i,JE} = 0, \quad (27-b)$$

giving

$$\Delta \phi_{i,0} = \Delta \phi_{i,2} \quad (28-a)$$

$$\Delta \phi_{i,JE+1} = \Delta \phi_{i,JE-1}. \quad (28-b)$$

Applying these to equation (26) we obtain

$$b = 0,$$

$$a = 1 + \frac{\Delta t}{g_2^2},$$

$$c = -\frac{\Delta t}{g_2^2} \quad \text{at } j=1,$$

and

$$b = -\frac{\Delta t}{g_{JE}^2},$$

$$a = 1 + \frac{\Delta t}{g_{JE}^2},$$

$$c = 0 \quad \text{at } j=JE.$$

The system of equations on the i th vertical line is in a scalar tridiagonal form and can be solved efficiently.

To summarize, in the ADI method we can split our solution scheme into two steps. One in the x -direction and one in the y -direction. After solving the block-tridiagonal systems in horizontal sweeps through the whole region, we make use of the intermediate increment $\Delta\phi^*$ to solve the tridiagonal system of equations in vertical sweeps. Adding the resulting $\Delta\phi^n$ to our original ϕ^n we finish one time-step forward. This process is repeated till it converges. This scheme is much faster than the successive horizontal line relaxation method of the previous chapter. It is further speeded up by using variable Δt 's cyclically. It is about three to five times faster than the SLR scheme depending on the Mach number, as discussed in Chapter V.

CHAPTER IV

MULTIPLE GRID METHOD

In the multiple grid method, a sequence of grids with geometrically decreasing mesh sizes are used to accelerate the convergence of a relaxation algorithm. It was first proposed by Federenko [12] subsequently analyzed by Bakhvalov [13] and Brandt [14,15] and applied to transonic flow problems by Jameson [1], and Brandt and South [2].

Suppose there is a sequence of grids $G^1, G^2, \dots, G^M$ which discretize a domain Ω with increasing mesh sizes $h^k = 2^{k-1}h^1$. That is, the grid-lines of G^k are every other grid-lines of G^{k-1} (Actually this is not necessary, but makes it convenient). Consider a linear differential equation

$$LU = F \quad \text{in } \Omega. \quad (29)$$

The difference approximation on each grid level G^k is of the form

$$L^k U^k = F^k \quad \text{in } \Omega \quad (30)$$

where U^k is the exact solution of the difference equation.

Let u^k be an approximation of the solution to U^k and v^k be the correction to U^k such that $U^k = u^k + v^k$ satisfying the difference equation (30). Since L is a linear operator, equation (30) can be replaced by

$$L^k v^k = f^k \quad (31)$$

$$\text{where } f^k = F^k - L^k u^k \quad (32)$$

is the residue left by u^k in satisfying the equation (30). Then the equation to be satisfied by the correction function on the coarser grid G^{k+1} is approximated by

$$L^{k+1} v^{k+1} = f^{k+1} = I_k^{k+1} f^k \quad (33)$$

where I_k^{k+1} is an injection operator which represents either a straight injection of values of f^k to the coarse grid G^{k+1} from the corresponding points on the fine grid G^k or a transfer of a weighted average of values of f^k on neighboring points. Once the equation (33) is solved, its solution is used as an approximate correction to u^k

$$u_{\text{new}}^k = u_{\text{old}}^k + I_{k+1}^k v^{k+1} \quad (34)$$

where I_{k+1}^k is an interpolation operator from the coarse grid G^{k+1} to the fine grid G^k . The same procedure can be

used for solving the equation(33) using a yet coarser grid G^{k+2} . Thus a sequence of grids can be used. This is the so called "Correction Storage (CS)" scheme for linear operator, in which one needs to store all corrections v^k on each grid level G^k in the fine-to-coarse process for future use in the coarse-to-fine process. The storage required for this algorithm is only a fraction more than the number of locations, n say, required to store corrections on the finest grid level. Since each level has $1/2^d$ as many grids as the last finer grid level for a d -dimensional problem, the total required locations for corrections are

$$(1+2^{-d}+2^{-2d}+ \dots)n < (2^d/2^d - 1)n < 2n \quad . \quad (35)$$

Hence, for a 2-D flow case, only $n/3$ more storage locations are required than the single grid method. The flow chart of the scheme is shown in Fig-5.

Generally for nonlinear problems we should store the full current approximation $\bar{u}^k$ in all grids instead of the corrections v^k . The corresponding scheme is as follows. Equation (30) can be rewritten as

$$L^k u^k = F^k = f^k + L^k u^k \quad (36)$$

where $f^k = F^k - L^k u^k$. Let $\bar{u}^k$ be an approximate solution of

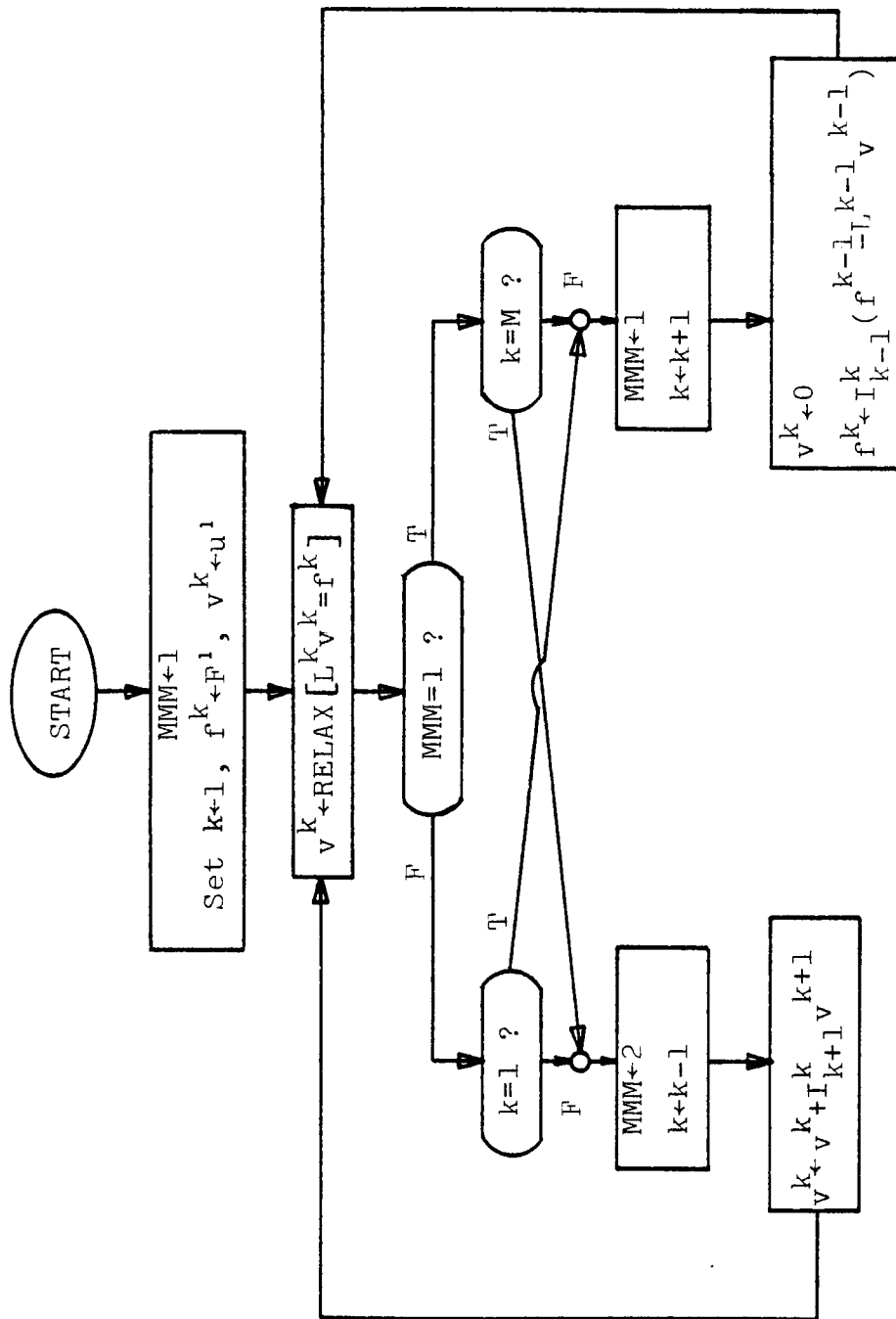


Figure 5. Correction Storage (CS) Algorithm.

the equation (36) for U^k . We are going to find improvements to $\bar{u}^k$ by going to a coarser grid G^{k+1} and later coming back to the grid G^k . Since the operator L^{k+1} in the coarser grid G^{k+1} will have more truncation error than the operator L^k in grid G^k , we approximately correct this additional error as we approximate the equation (36) in the coarser grid G^{k+1} as follows. An approximation to the equation (36) on the coarser grid G^{k+1} is of the form

$$L^{k+1} \bar{u}^{k+1} = \bar{f}^{k+1} \quad (37)$$

where $\bar{f}^{k+1} = I_k^{k+1} f^k + L^{k+1} I_k^{k+1} \bar{u}^k$. In the above approximation, the term $\bar{f}^{k+1}$ can be interpreted as follows:

$$\begin{aligned} \bar{f}^{k+1} &= I_k^{k+1} (\bar{f}^k - L^k \bar{u}^k) + L^{k+1} I_k^{k+1} \bar{u}^k \\ &= I_k^{k+1} \bar{f}^k + (L^{k+1} I_k^{k+1} \bar{u}^k - I_k^{k+1} L^k \bar{u}^k) \end{aligned} \quad (38)$$

The last term on the right is the compensating term for the difference in the truncation errors of the operators L^k and L^{k+1} .

When the solution $\bar{u}^{k+1}$ to the equation (37) is obtained on the coarser grid G^{k+1} , we can go back to the finer grid G^k to find an improved solution for $\bar{u}^k$ by

$$\bar{u}_{\text{new}}^k = \bar{u}_{\text{old}}^k + I_{k+1}^k (\bar{u}^{k+1} - I_k^{k+1} \bar{u}_{\text{old}}^k) \quad (39)$$

where $(\bar{u}^{k+1} - I_k^{k+1} \bar{u}_{\text{old}}^k)$ is the correction term and note that $I_{k+1}^k I_k^{k+1} \bar{u}^k \neq \bar{u}^k$ generally. This is the "Full Approximation Storage (FAS)" algorithm.

The FAS mode has the same flow chart as the CS algorithm as shown in Fig.5 except that the equations (33), (32), and (34) are replaced by (37), (38), and (39), respectively. It is flow-charted in Fig-6.

Acceleration in convergence of the multiple grid method is based on two reasons:

- (1) Only high-frequency components of errors are needed to be reduced on each level; therefore, the convergence rate is faster.
- (2) The number of mesh points on the coarse-grid-level is much less than that on the fine-grid-level; therefore, the required number of operations decreases.

In effect, high frequency components of the errors are dissipated on fine grid and lower frequency components are reduced on the coarse grids, thus speeding up the overall convergence of the iterations.

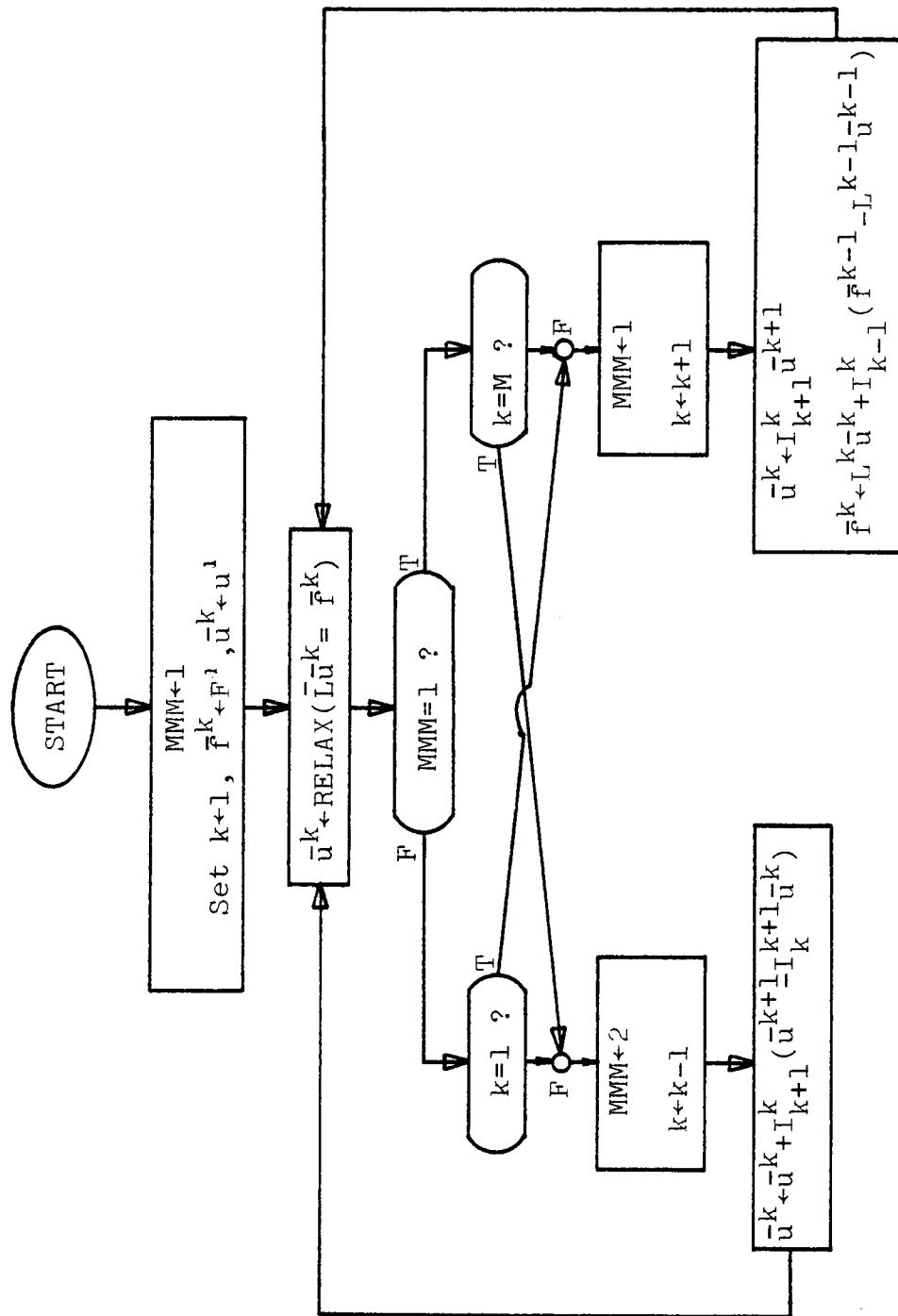


Figure 6. Full Approximation Storage (FAS) Algorithm.

CHAPTER V

RESULTS AND CONCLUSIONS

All the calculations are done for a flow past a symmetric circular airfoil with thickness ratio $1/6$ in a rectangular domain under various conditions. Two kinds of flow conditions are studied: subcritical case ($M_\infty=0.696$) and supercritical case ($M_\infty=0.791$). In the subcritical case the flow is in a wind tunnel with normal velocity boundary condition imposed at the wall. In the supercritical case we have considered two different domains: one a semi-infinite domain with free flow far field boundary conditions and another a confined domain in a wind tunnel. The domains in all cases are discretized into a 129×33 grid except for the semi-infinite domain case in which it is a 129×41 grid. In the semi-infinite domain case, for the upper boundary condition normal velocity is set to zero to simulate the free flow. In other cases normal velocity distributions as measured in wind tunnel tests are imposed at the upper boundary.

The C_p distributions of the fully converged results by all relaxation schemes are shown in Figures 7, 10, and 13, for various flow conditions. The convergence rates are compared by plotting the L_1 norms of residue in Figures 8, 11, and 14, and the L_1 norms of $\Delta\phi$ in Figures 9, 12, and 15 against the work units. The work units are measured by the equivalent number of sweeps on the finest grid. Therefore the work units are just equal to the number of sweeps for the SLR method and the ADI method. In the multi-grid method each level has about $1/4$ as many grid points as the next finer grid level. Thus the work needed for each sweep on G^k is only $1/4$ of the next finer level G^{k-1} .

The L_1 norm is calculated by

$$||Z|| = \left(\sum_{i=1}^{JEND} \sum_{j=1}^{JEND} |Z_{i,j}| \Delta A_{i,j} \right) / A$$

where $Z_{i,j}$ can be either the residue or $\Delta\phi$ at point (i,j) , $\Delta A_{i,j}$ is the cell area at (i,j) , and A is the total area of the domain.

The inviscid-transonic similarity parameter K_i is calculated by equation (2) and the viscous-transonic similarity parameter K_v is set to be 0.01. In multigrid method the K_v on G^k is 4 times the K_v on the next finer grid G^{k-1} , but ultimately the converged results correspond to the value of K_v (0.01) on the finest grid.

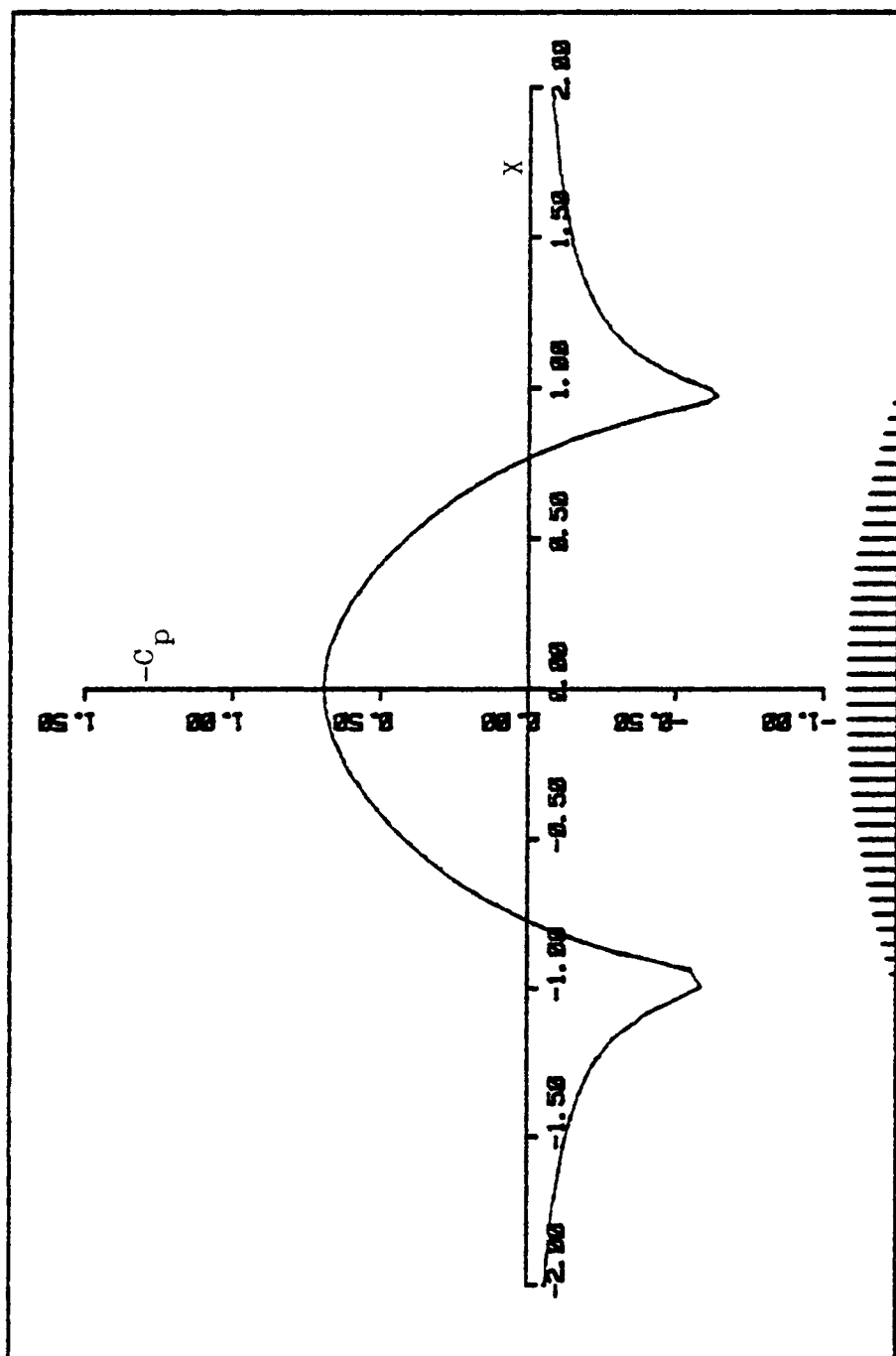
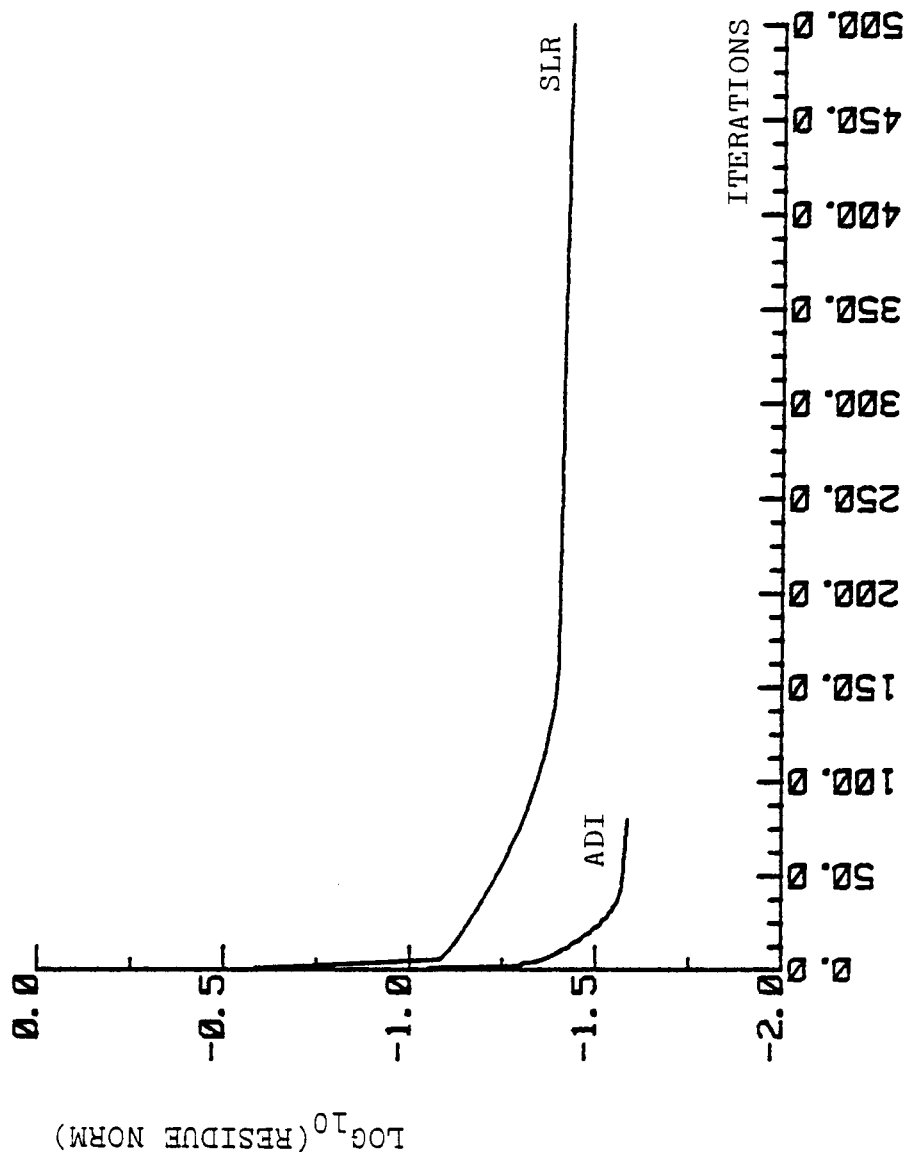


Figure 7. Surface Pressure Coefficient on a 1/6 Thick Circular Airfoil in a Wind Tunnel for $M_\infty = 0.696$.



(a) SLR and ADI Methods

Figure 8. Convergence History of Residue for $M_{\infty}=0.696$ in a Wind Tunnel.

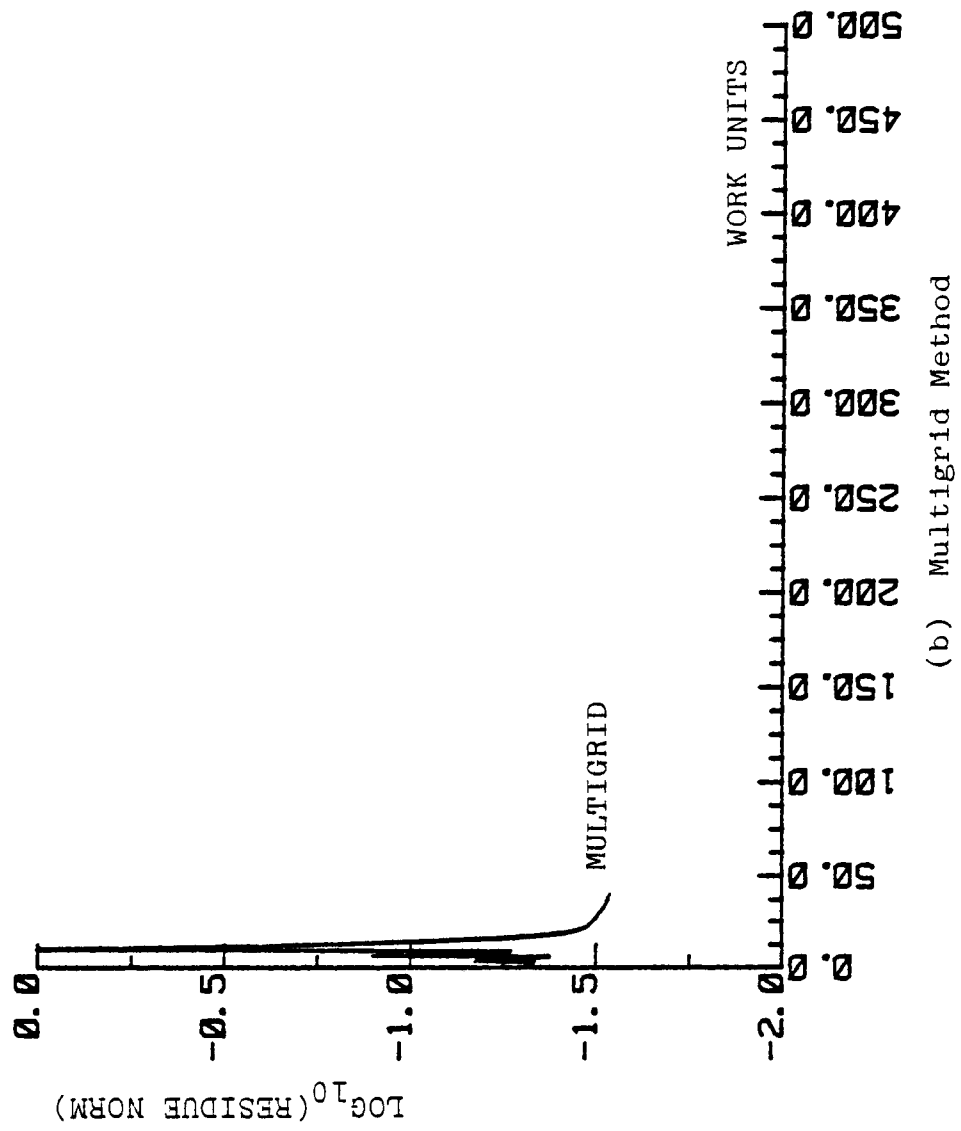


Figure 8.(continued)

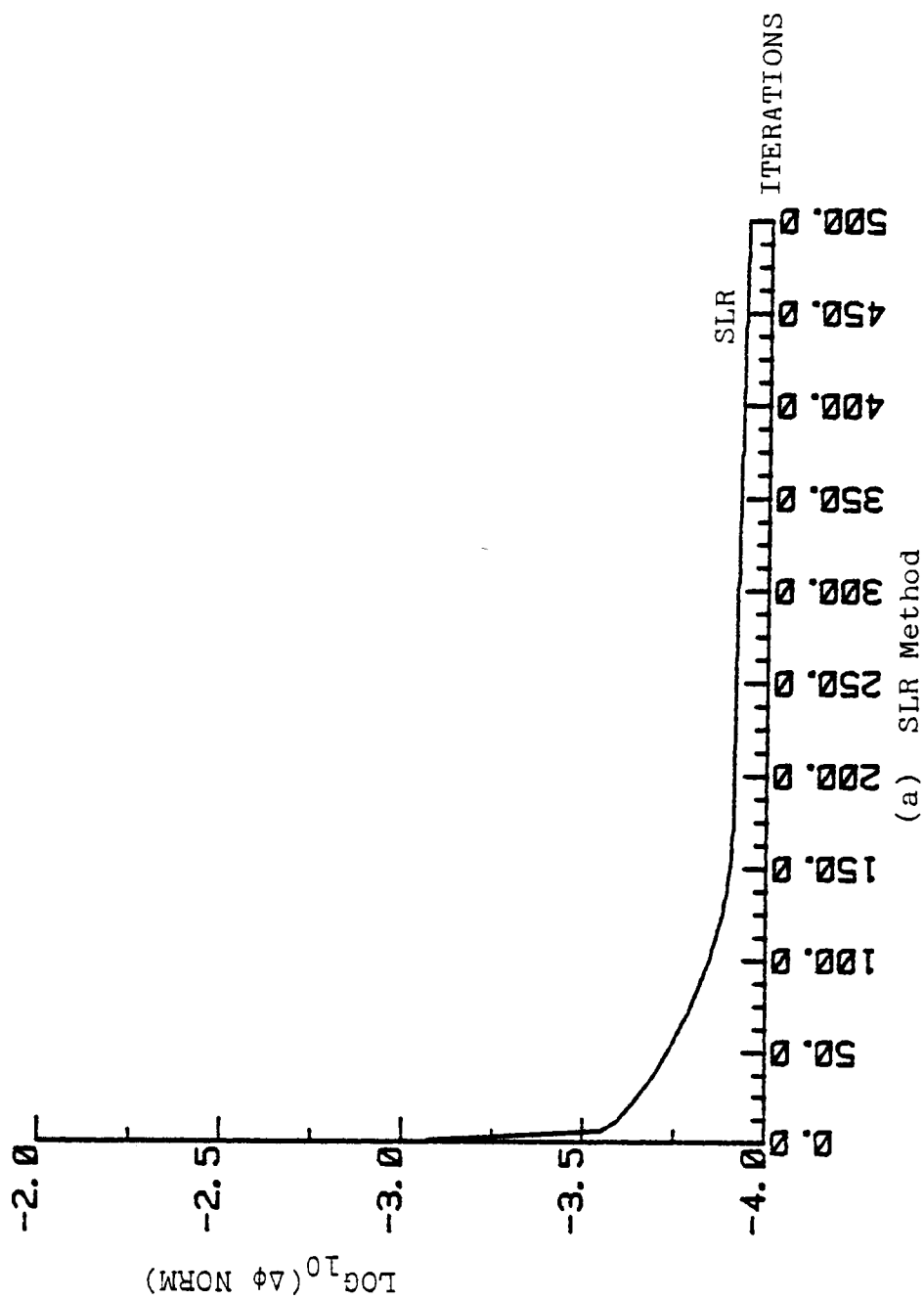


Figure 9. Convergence History of $\Delta\phi$ for $M_\infty=0.696$ in a Wind Tunnel.

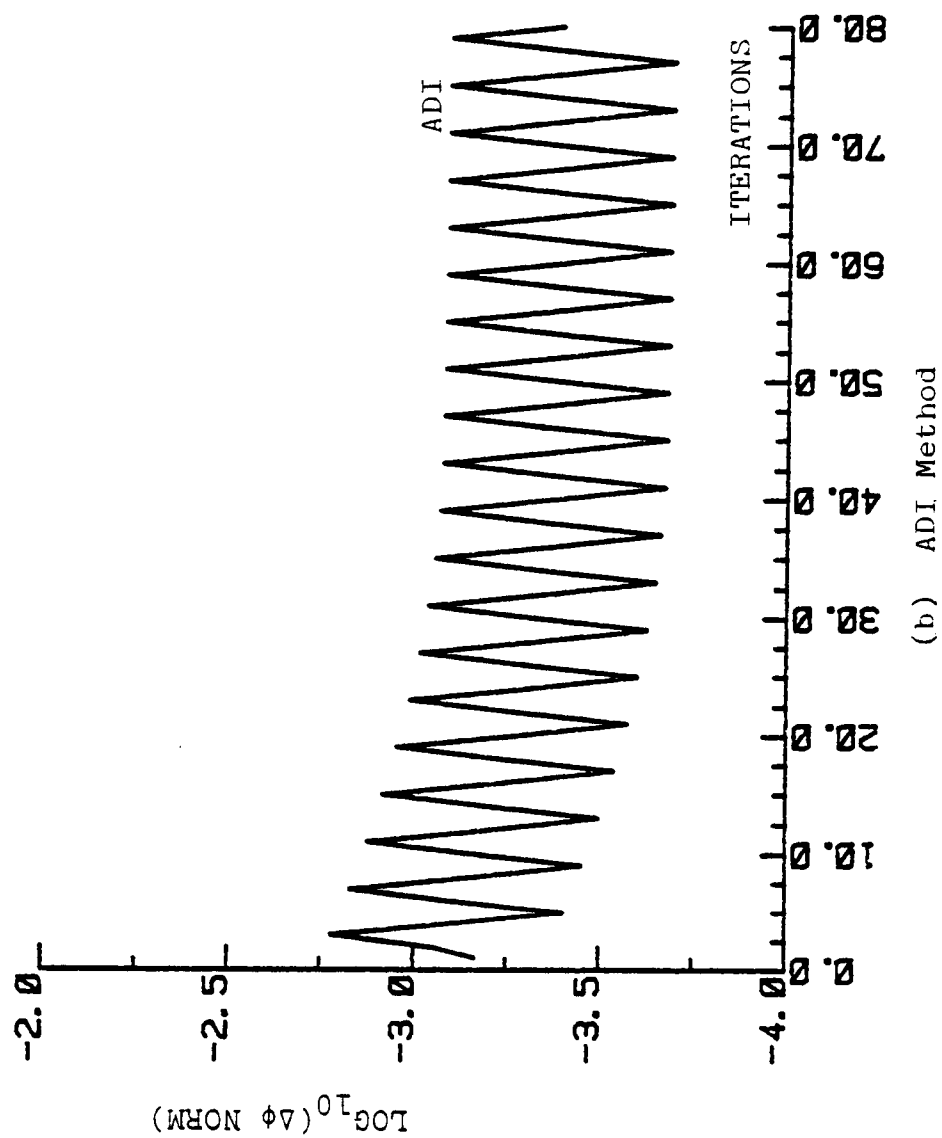
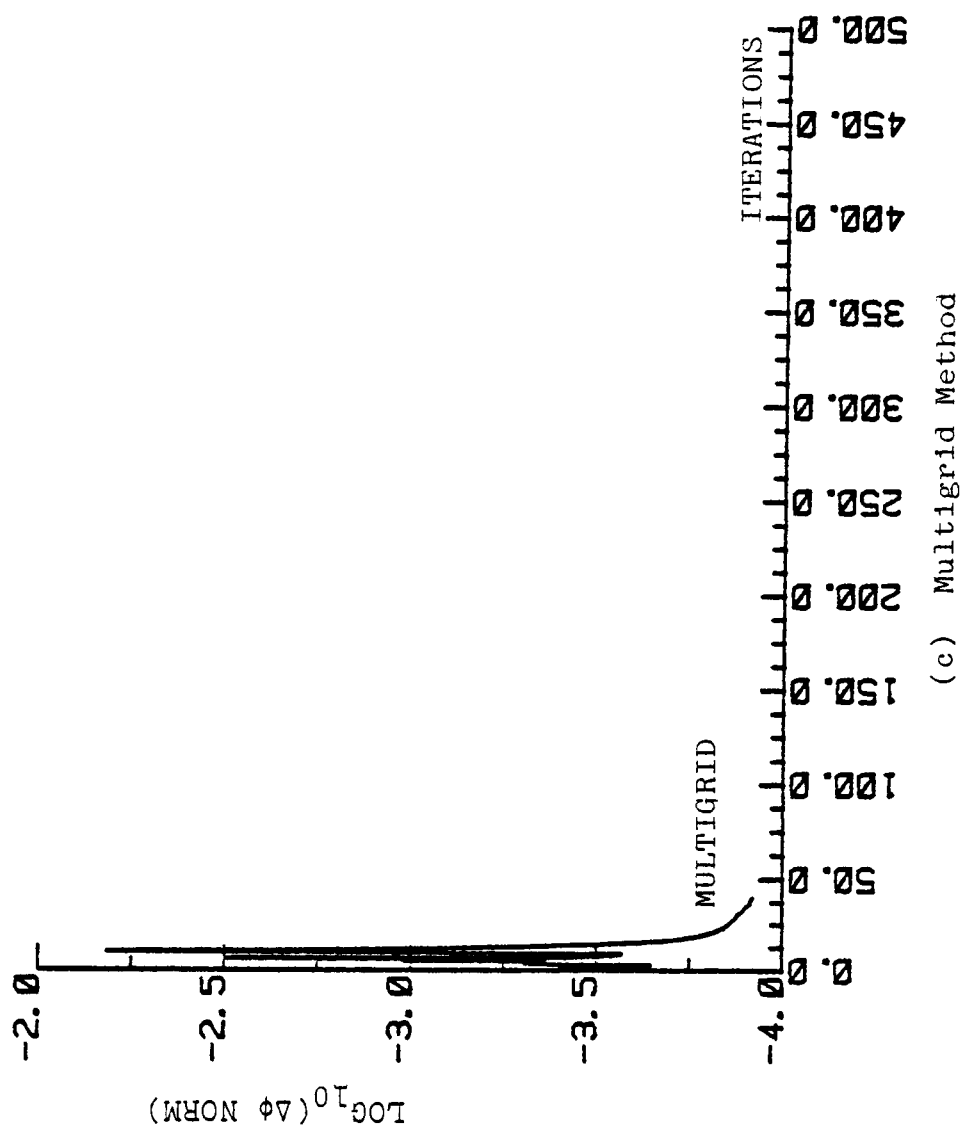


Figure 9: (continued)



(c) Multigrid Method

Figure 9. (continued)

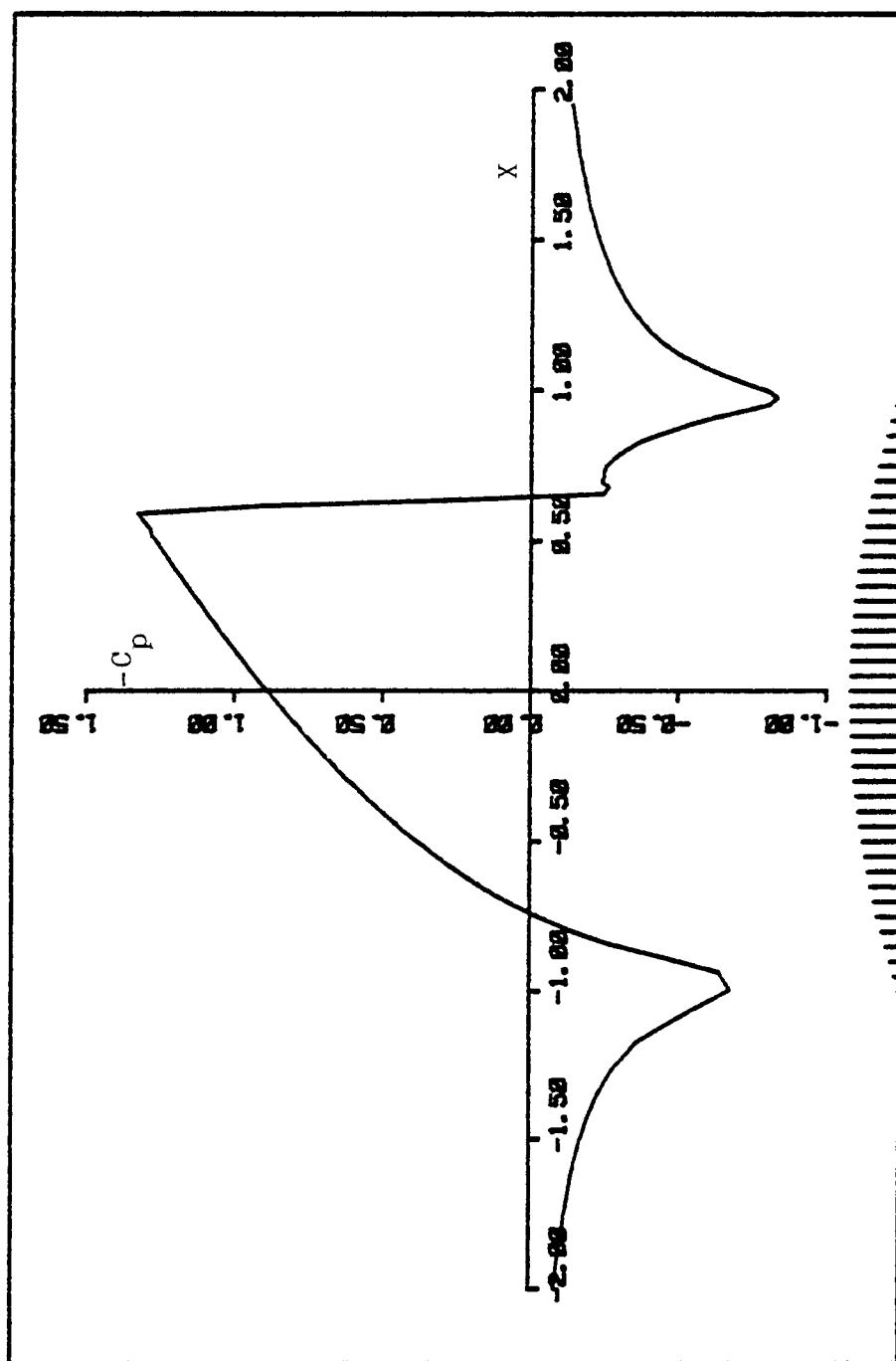


Figure 10. Surface Pressure Coefficient on a 1/6 Thick Circular Airfoil in a Wind Tunnel for $M_\infty = 0.791$.

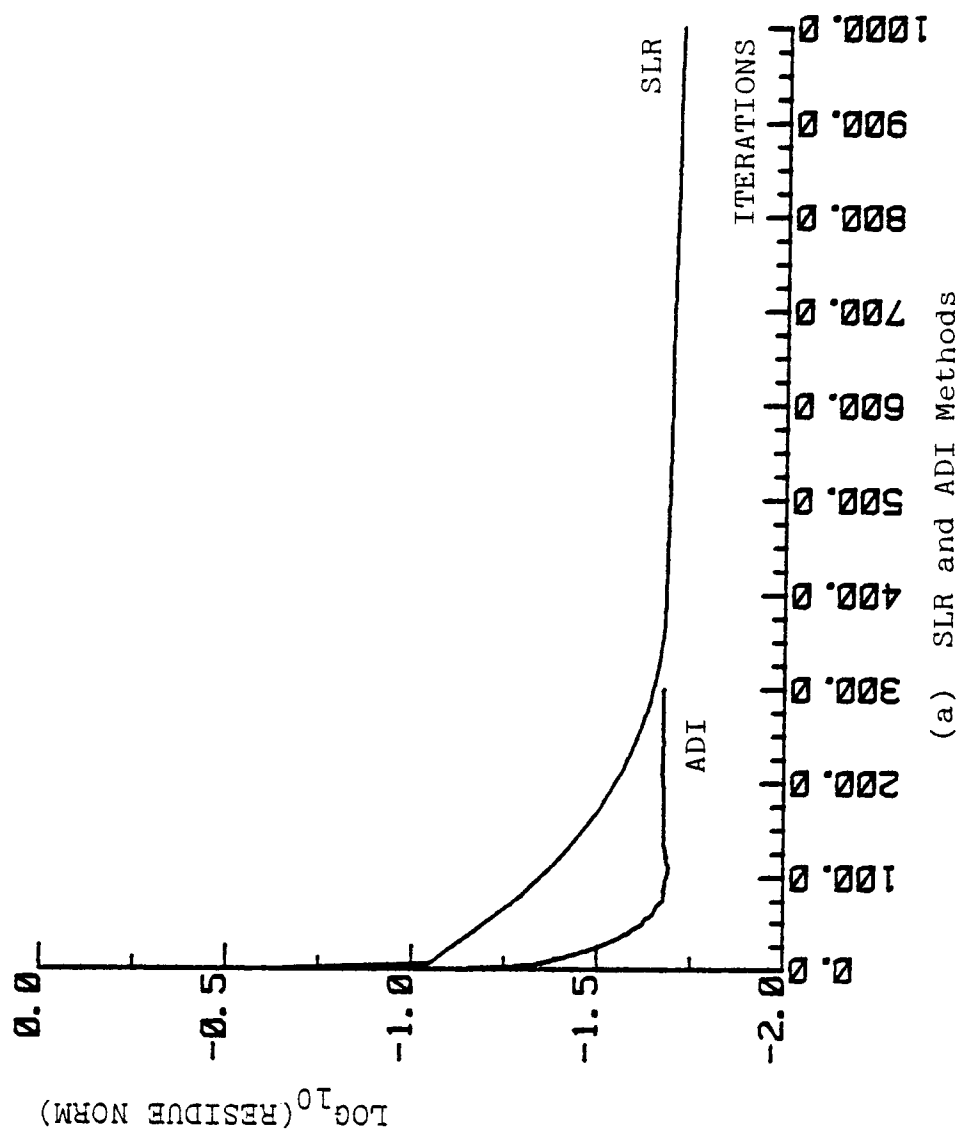


Figure 11. Convergence History of Residue for $M_{\infty}=0.791$ in a Wind Tunnel.

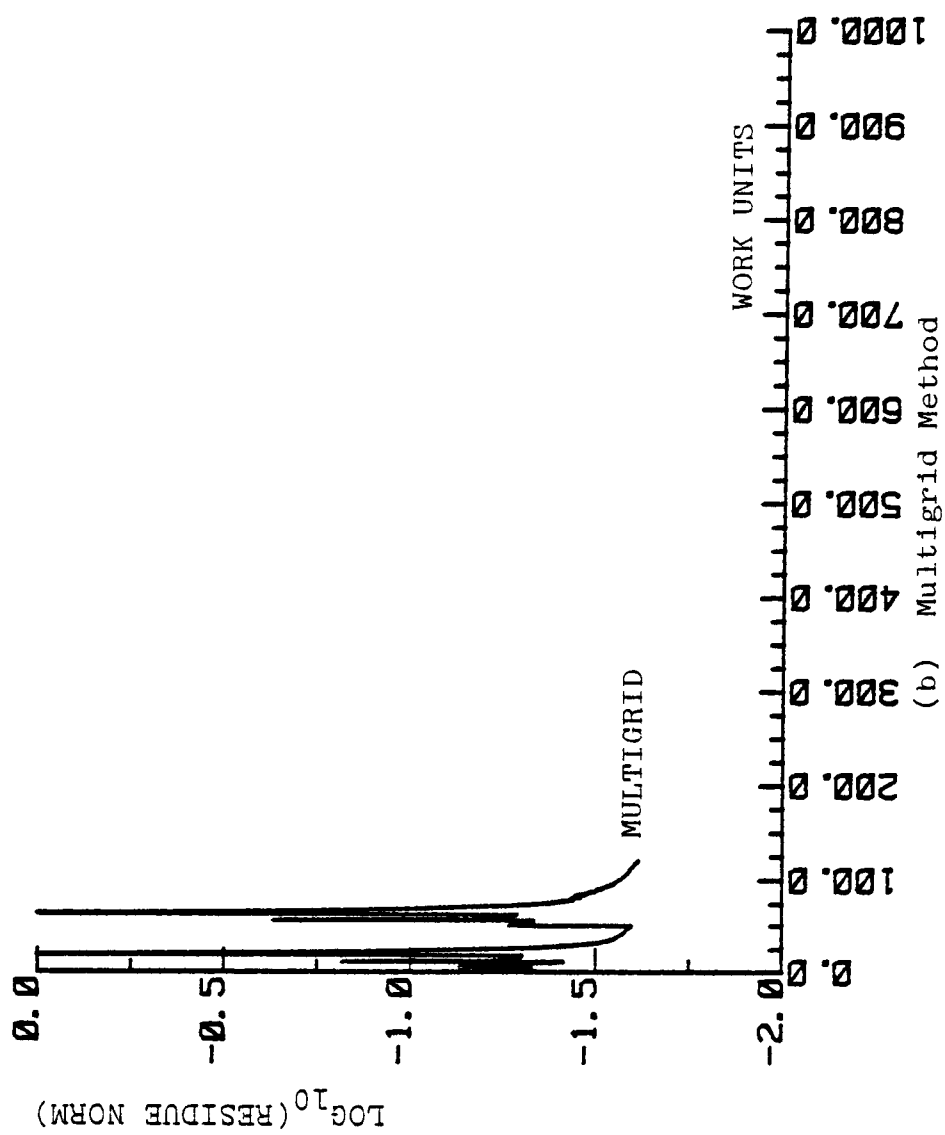


Figure 11. (continued)

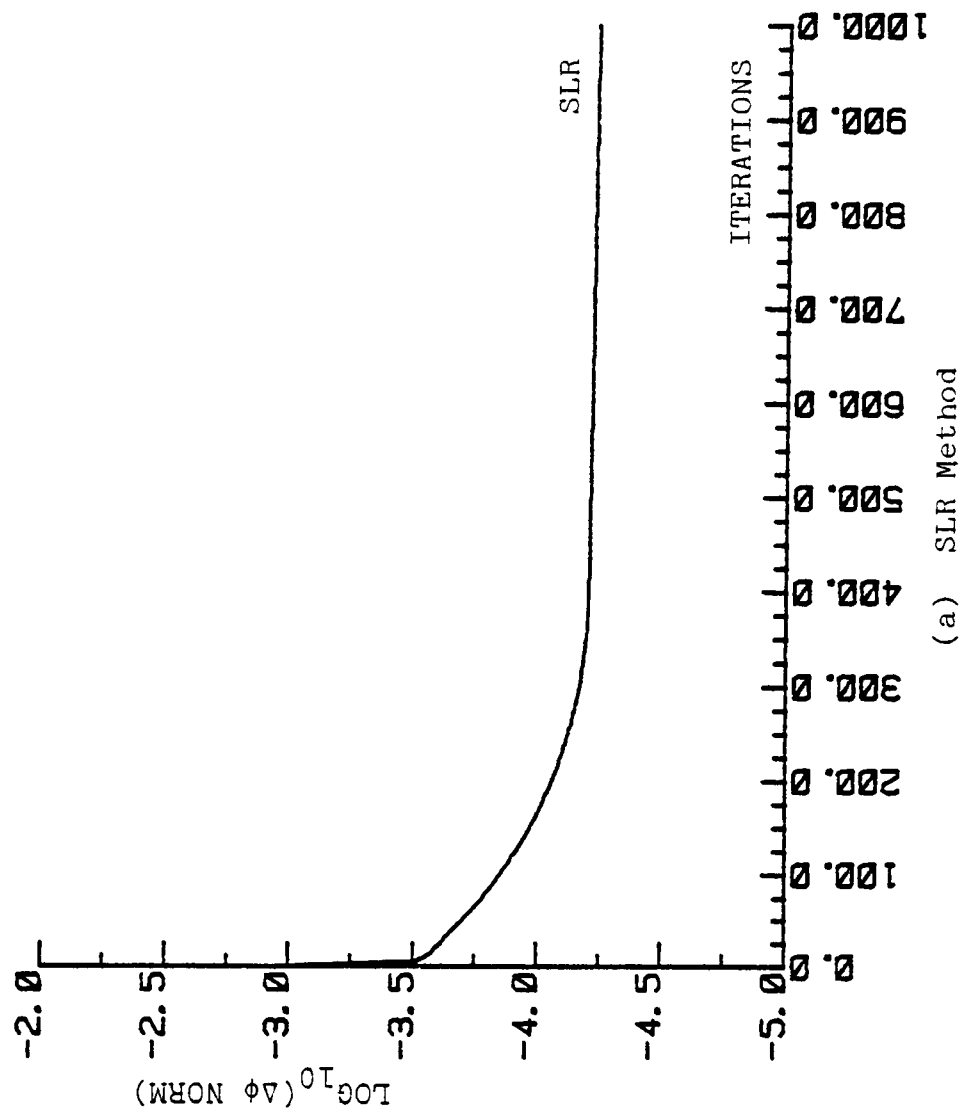


Figure 12. Convergence History of $\Delta\phi$ for $M_\infty=0.791$ in a Wind Tunnel.

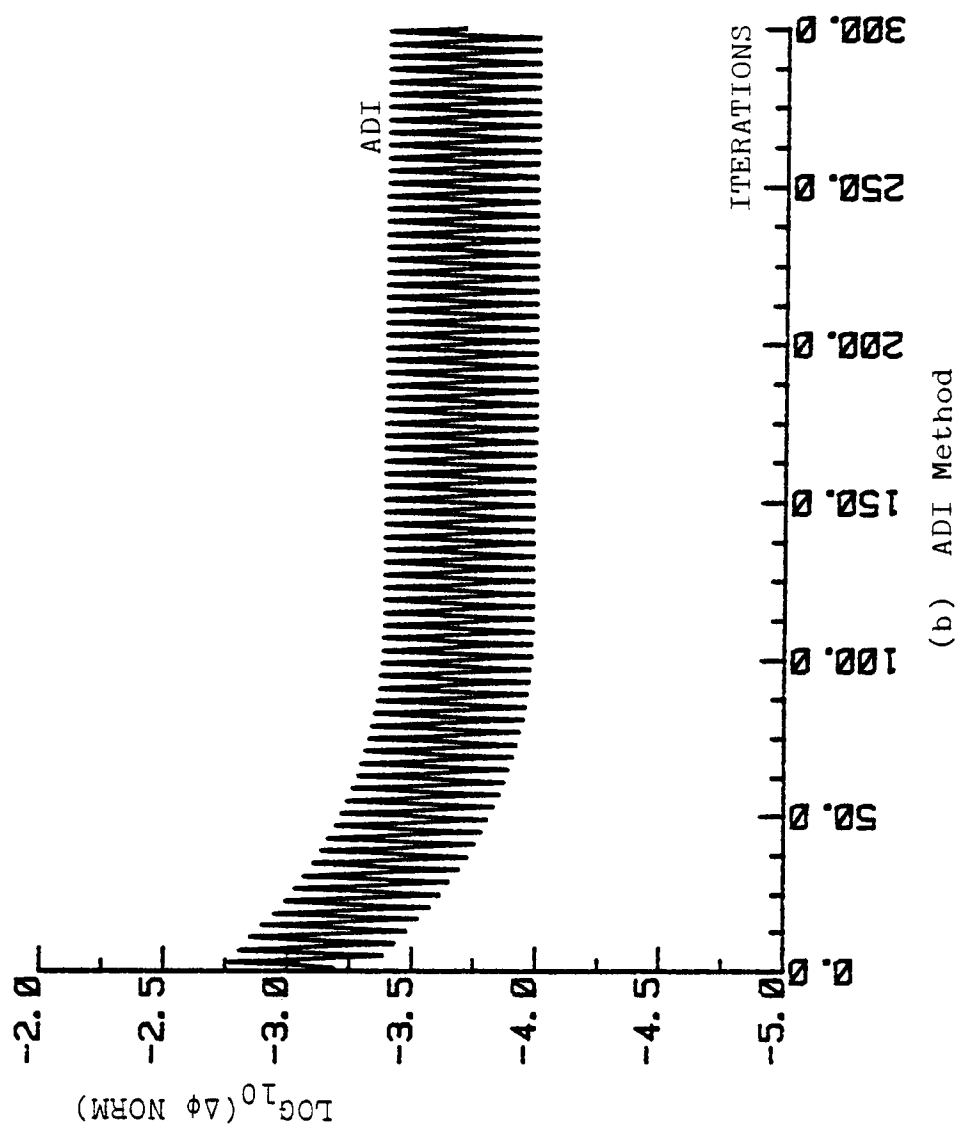


Figure 12. (continued)

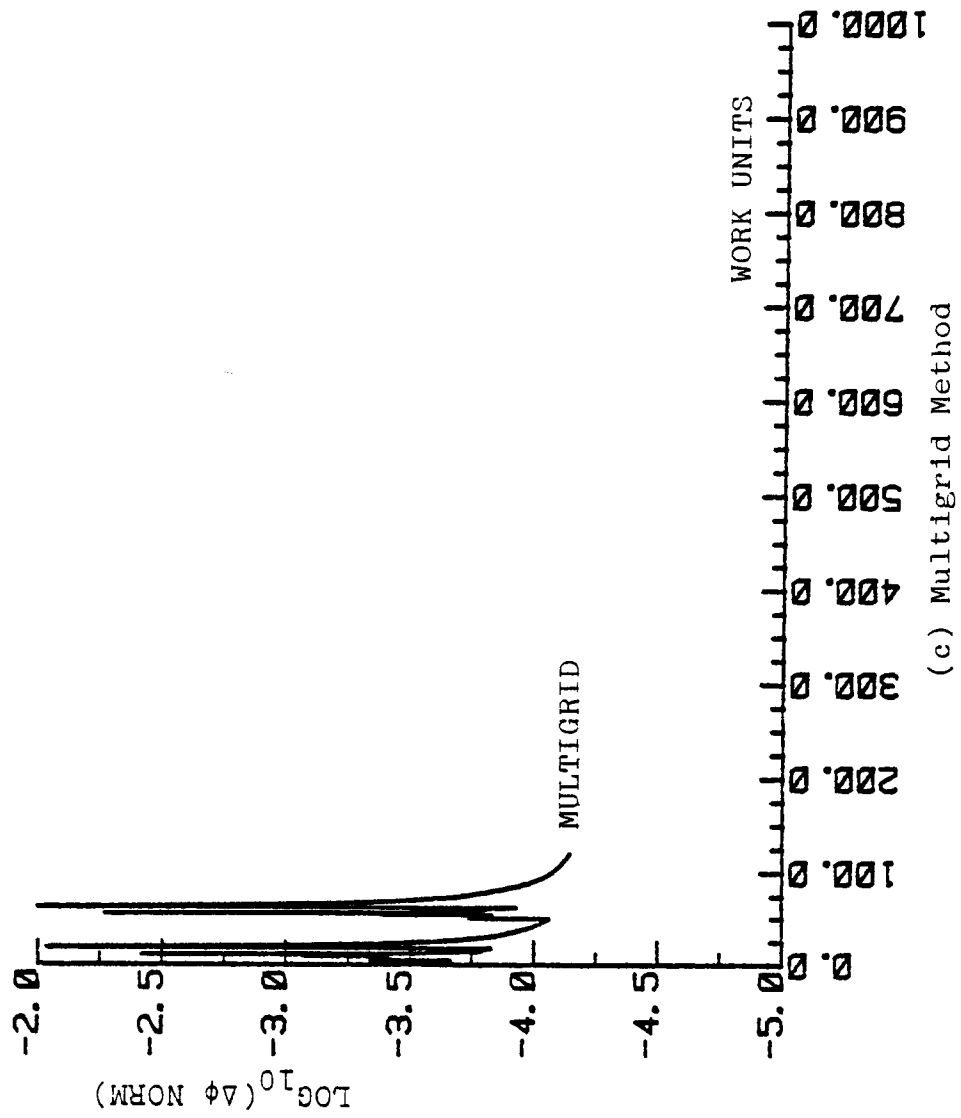


Figure 12.(continued)

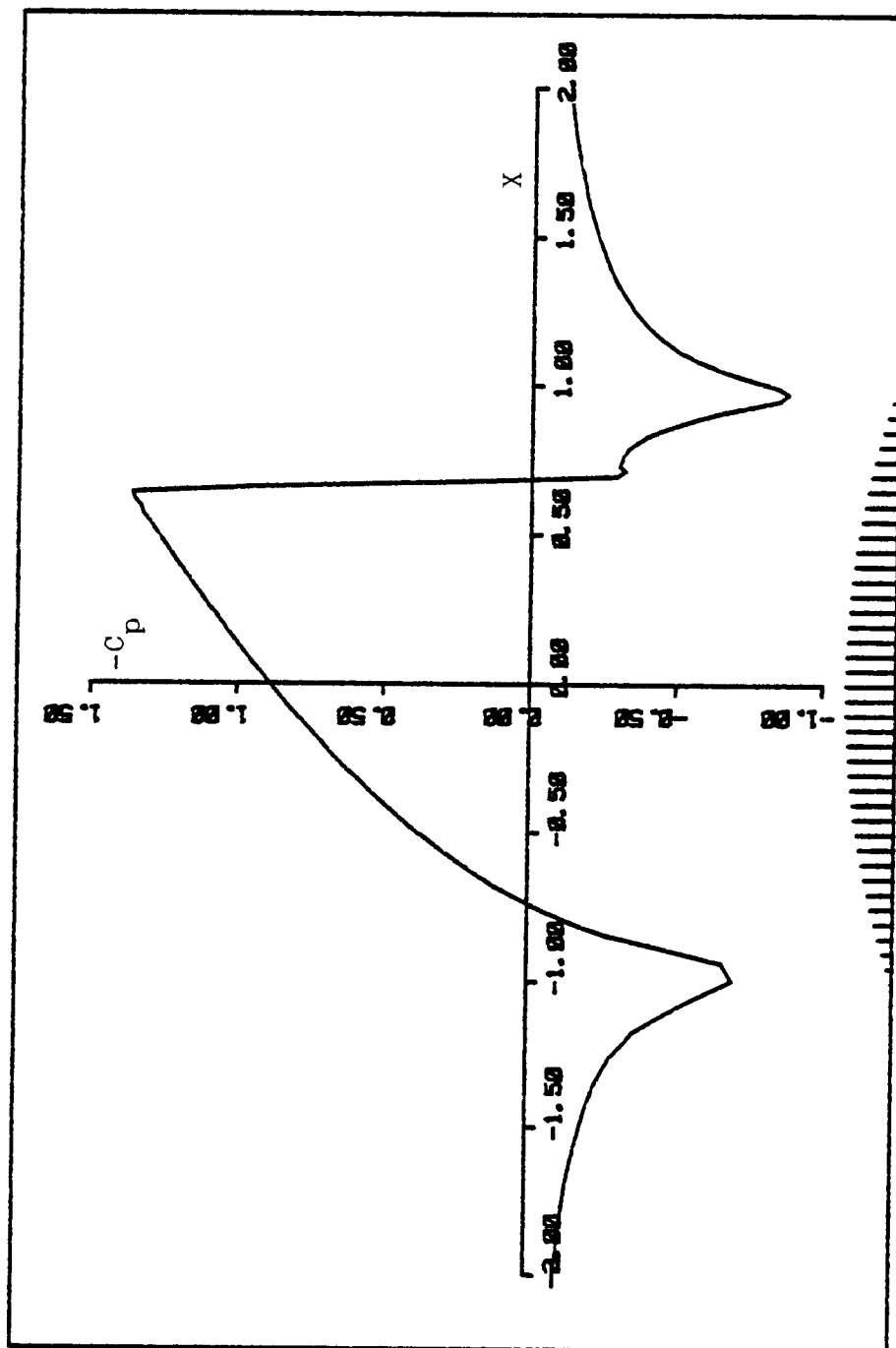


Figure 13. Surface Pressure Coefficient on a $1/6$ Thick Circular Airfoil in a Semi-Infinite Domain for $M_\infty = 0.791$.

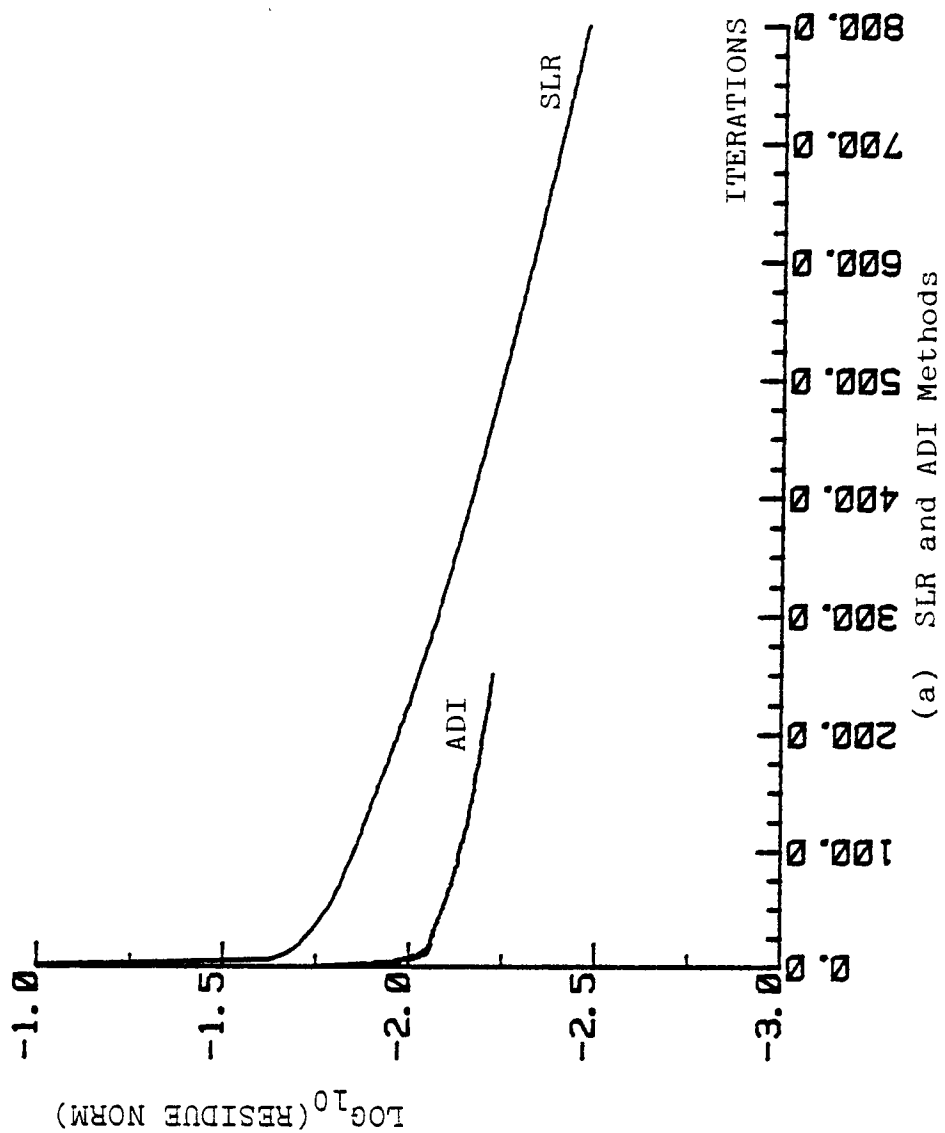
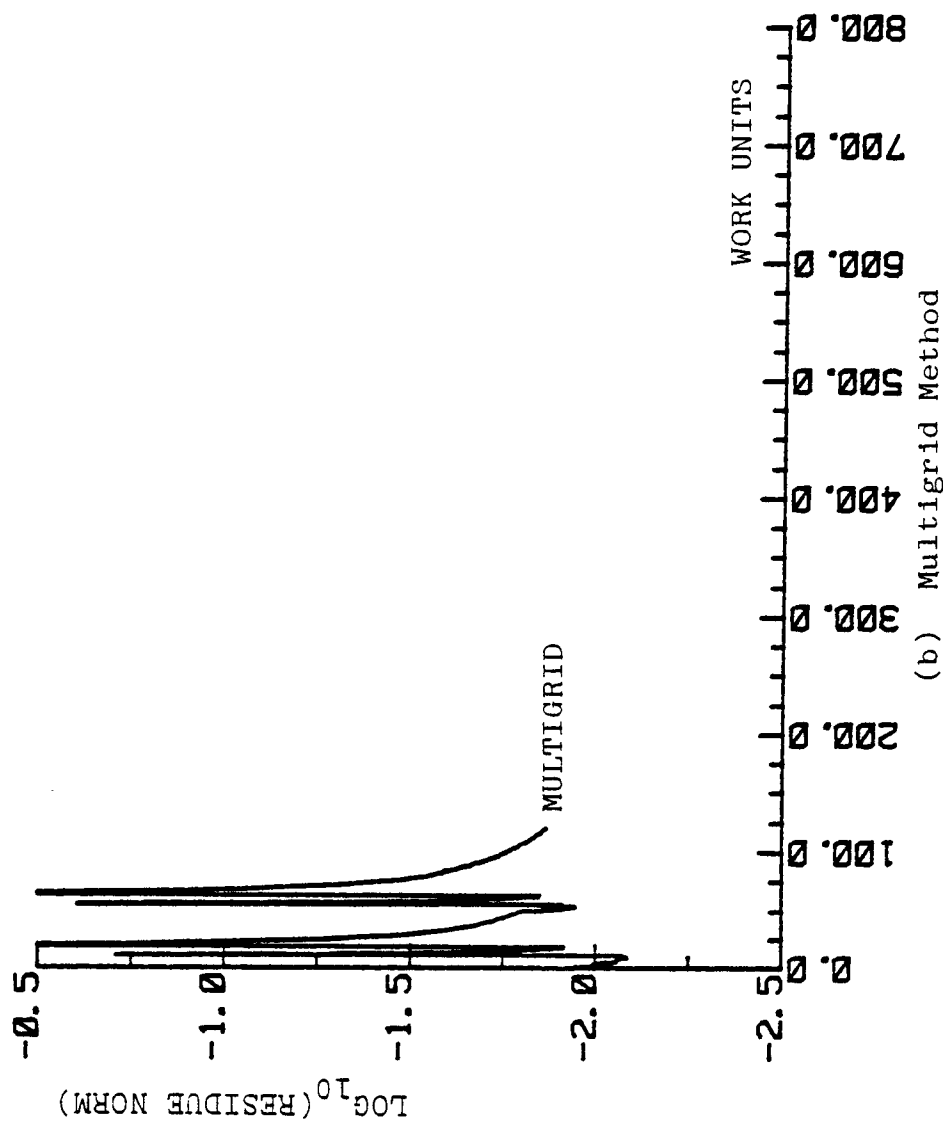
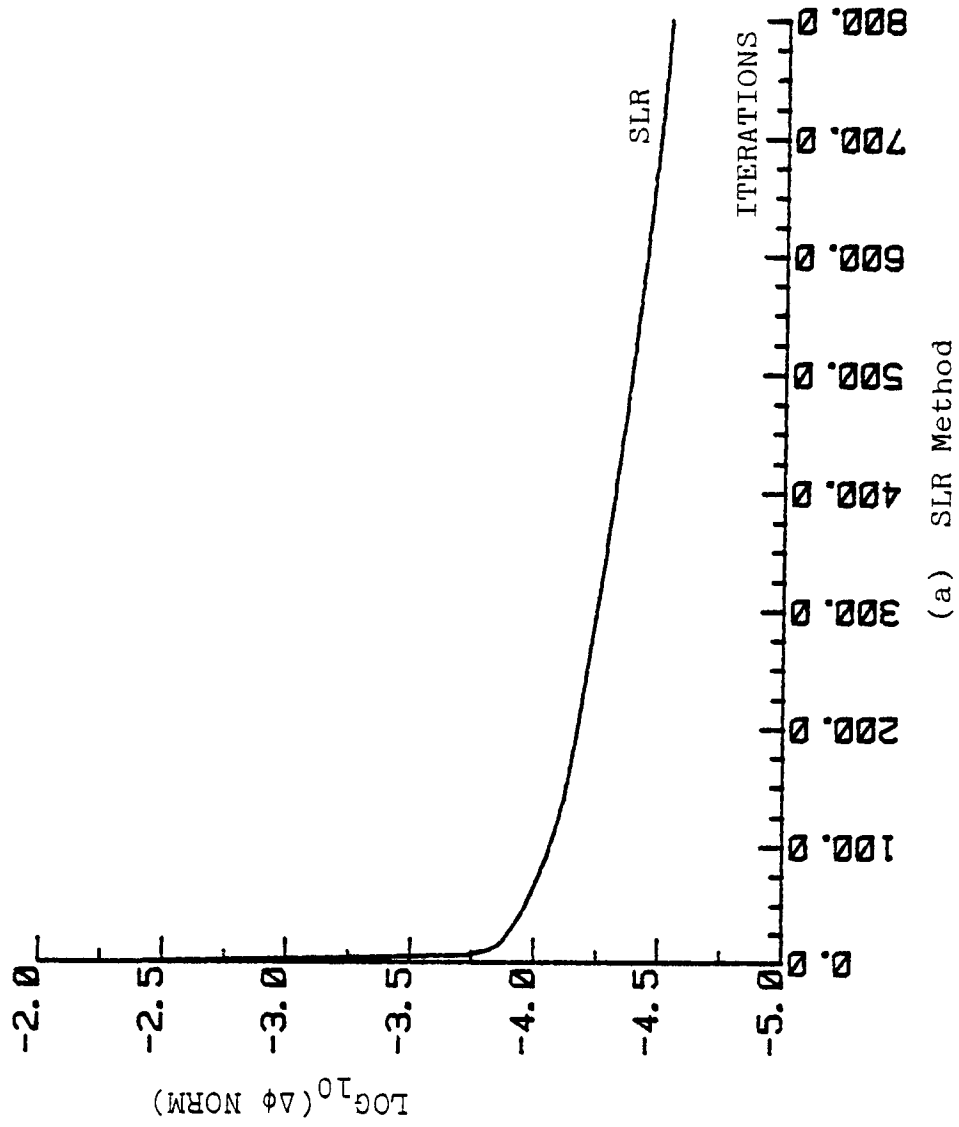


Figure 14. Convergence History of Residue for $M_\infty = 0.791$ in a Semi-Infinite Domain.



(b) Multigrid Method

Figure 14. (continued)



(a) SLR Method

Figure 15. Convergence History of $\Delta\phi$ for $M_\infty = 0.791$ in a Semi-Infinite Domain.

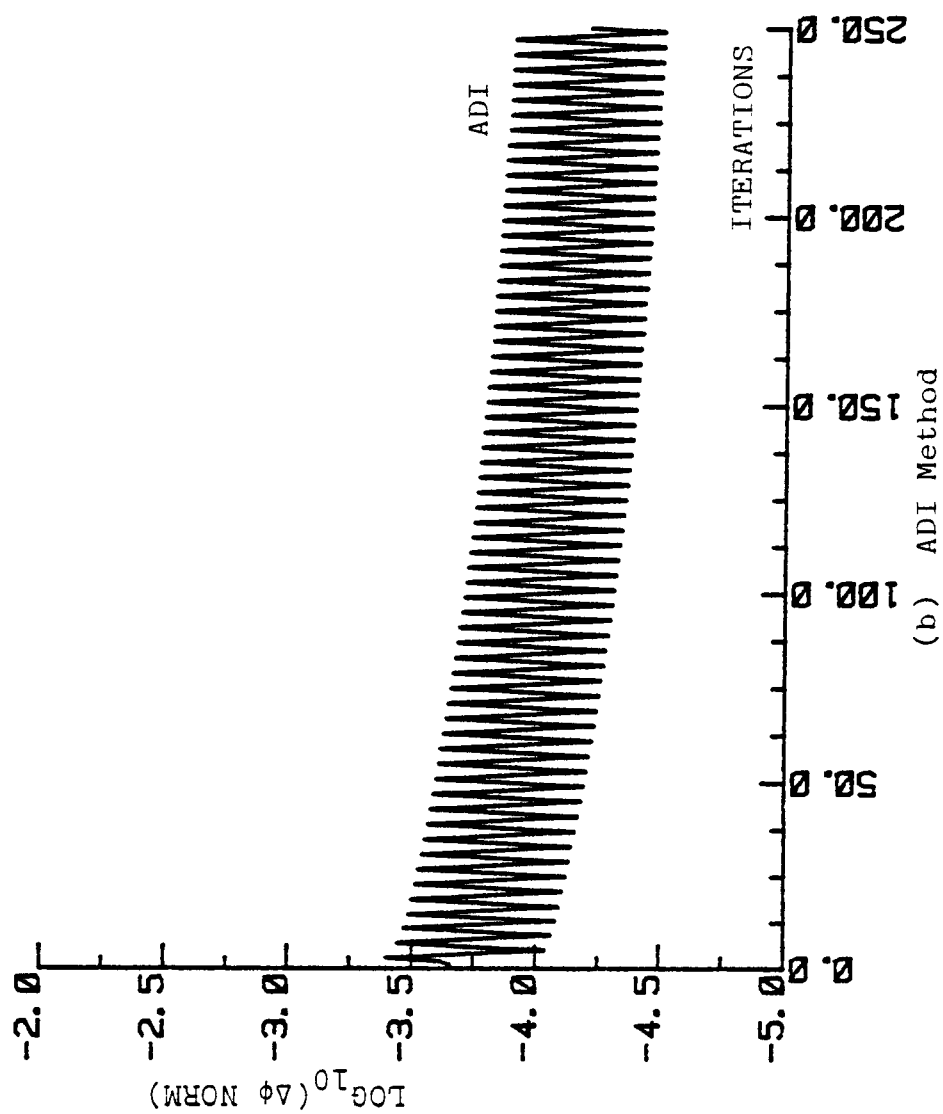
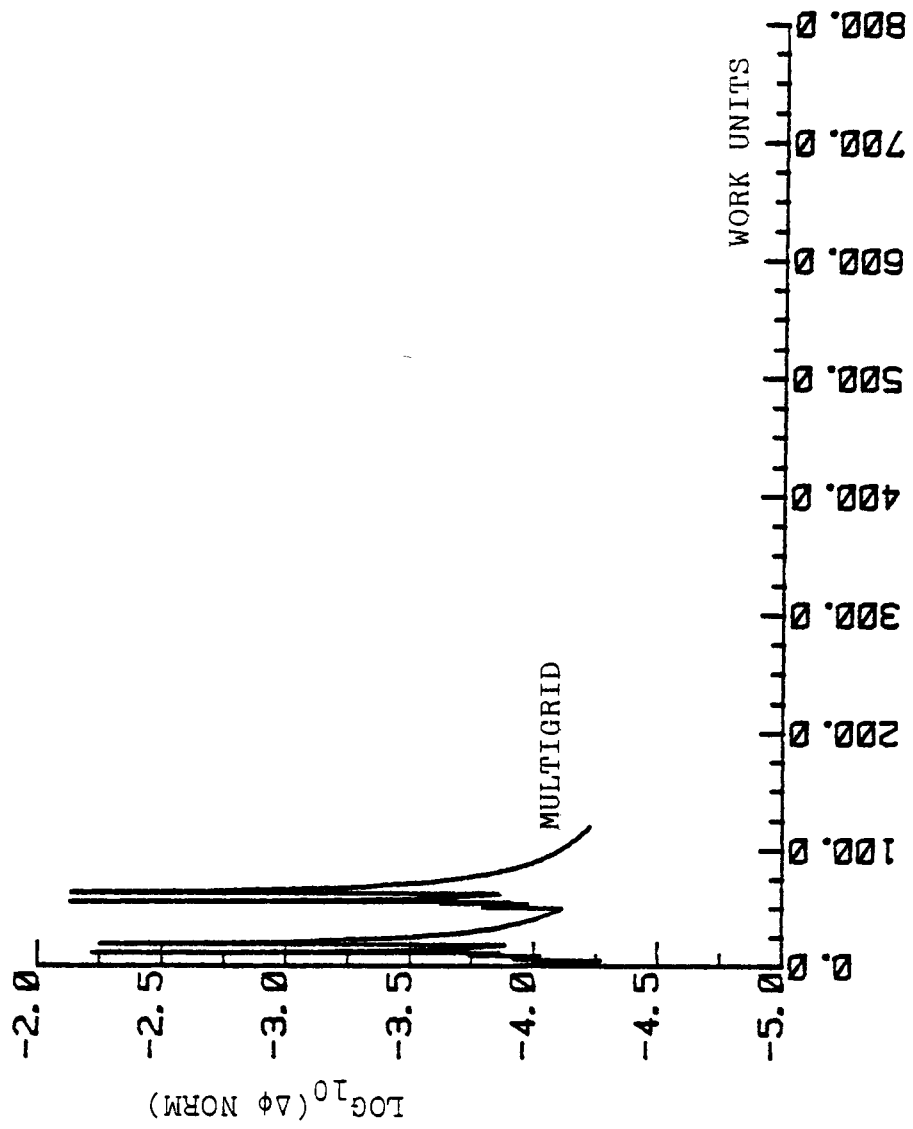


Figure 15.(continued)



(c) Multigrid Method

Figure 15.(continued)

The time-increment Δt in the straight ADI method can be a fixed small value. It has been shown that the convergence rate of the ADI method can be improved by a sequence of Δt s. In our case we adopt 0.01, 0.02, 0.04, 0.02 cyclically. The convergence rate is much faster than that for fixed Δt . In multigrid method Δt is fixed on each level G^k and is 2 times that of the next finer grid G^{k-1} .

From the residue-work plots it can be seen the ADI method is about 3 times faster than the SLR method and multiple grid method is about 2 times faster than the ADI method. In purely subcritical flows the ADI method is 5 times faster than the SLR.

The following conclusions may be drawn from the numerical schemes presented in this thesis.

- (i) The small disturbance transonic equation can be solved by second order accurate central difference approximations, by considering the compressive viscosity term like an artificial viscosity.
- (ii) Successive line relaxation (SLR) methods are successful in solving the non-linear difference equations. But the convergence rate can be speeded up by a factor of 3 to 5 by the use of alternating direction implicit methods. The speed up is less in

supercritical flows than in the subcritical flows. Further work is needed to introduce acceleration parameters and optimize them to speed up the ADI method further.

- (iii) Convergence rate of the ADI method is speeded up by a factor of 2 or more by the use of multiple grid scheme. However more work is required to make this scheme faster and more reliable.

BIBLIOGRAPHY

BIBLIOGRAPHY

1. A. Jameson, "Acceleration of Transonic Potential Flow Calculations on Arbitrary Meshes by the Multiple Grid," AIAA Paper, 79-1458, July 1979.
2. J. C. South, Jr. and A. Bandt, "Application of a Multi-Level Grid Method to Transonic Flow Calculations," Transonic Flow Problems in Turbomachinery, Hemisphere, Washington, 1976.
3. E. M. Murman and J.D. Cole, "Calculation of Plane Steady Transonic Flows," AIAA Journal, Vol. 9, No. 1, 1971, pp. 114-121.
4. E. M. Murman, "Analysis of Embedded Shock Waves Calculated by Relaxation Methods," Proceedings of AIAA Computational Fluid Dynamics Conference, 1973, pp. 27-40.
5. A. Jameson, "Transonic Potential Flow Calculations Using Conservation Form," Proceedings of AIAA Computational Fluid Dynamics Conference, June 1975, pp. 148-155.
6. M. Sichel, "Two-Dimensional Shock Structure," Advances in Applied Mechanics, Vol. 11, Academic Press, New York, 1971.
7. H. B. Keller, "A New Difference Scheme for Parabolic Problems," Numerical Solution of Partial Differential Equations - II, Academic Press, New York, 1970, pp. 327-350.
8. J. M. Wu et al., Fundamental Studies of Subsonic and Transonic Flow Separation, Part II - Second Phase Summary, Report AEDC-TR-75-95, Arnold Engineering Development Center, Tullahoma, TN, Sept. 1975.
9. D. W. Peaceman and H. H. Rachford, "The Numerical Solution of Parabolic and Elliptic Differential Equations," J. of Soc. Indust. App. Math., 1955, pp. 28-41.
10. W. F. Ballhaus, A. Jameson, and J. Albert, "Implicit Approximate - Factorization Schemes for the Solution of Steady Transonic Flow Problems," AIAA 3rd Computational Fluid Dynamics Conference, Albuquerque, New Mexico, June 1977.

11. R. F. Warming and R. M. Beam, "On the Construction and Application of Implicit Factored Schemes for Conservation Laws," SIAM-AMS Proceedings, Vol. 11, 1978, pp. 85-129.
12. R. P. Fedorenko, "A Relaxation Method for Solving Elliptic Difference Equations," USSR Computational Mathematical Physics, Vol. 1, pp. 1092-1096.
13. N. S. Bakhvalov, "On the Convergence of a Relaxation Method with Natural Constraints on the Elliptic Operator." USSR Computational Mathematics and Mathematical Physics, pp. 101-135.
14. A. Brandt, "Multi-Level Adaptive Computations in Fluid Dynamics," AIAA Paper, 79-1455, July 1979.
15. A. Brandt, "Multi-Level Adaptive Solutions to Boundary-Value Problems," Mathematics of Computation, Vol. 31, Number 138, April 1977, pp. 333-390.

VITA

Yeng-Yung Tsui was born in Hsinchu, Taiwan, Republic of China, on September 6, 1953. He graduated from the Provincial Hsinchu High School in 1971. After four years he received his Bachelor of Engineering in Mechanical Engineering from Tamkang College. Subsequently, he served as a Technical Second Lieutenant in the Chinese Army for two years. He joined the University of Tennessee Space Institute as a research assistant in the fall 1978. He received his degree of Master of Science in March 1981.